



Universidad
Carlos III de Madrid

Escuela Politécnica Superior UC3M

Departamento de Ingeniería Telemática

PROYECTO FIN DE CARRERA

Diseño de arquitectura y visualizaciones para un módulo de analítica del aprendizaje en **Open edX**

Autor: Javier Santofimia Ruiz

Tutor: Pedro José Muñoz Merino

Leganés, Septiembre de 2015

Título: Diseño de arquitectura y visualizaciones para un módulo de analítica del aprendizaje en Open edX

Autor: Javier Santofimia Ruiz

Director: Pedro José Muñoz Merino

EL TRIBUNAL

Presidente: _____

Vocal:

Secretario:

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día ____ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Agradezco a Pedro José Muñoz Merino, Héctor Javier Pijeira Díaz y José Antonio Ruipérez Valiente por el trabajo conjunto en el desarrollo del módulo que se describe en esta memoria.

Agradezco también a Manuela Ruiz Romero por la ayuda prestada en el formato y presentación final de este documento.



Resumen

Los cursos MOOC (Massive Online Open Course) están emergiendo con fuerza como un tipo importante de cursos online. Entre las características más importantes de estos cursos, se encuentra el carácter masivo de estos, suponiendo la generación de una gran cantidad de datos, creados a partir del registro de las interacciones de los usuarios con el curso. Interpretar correctamente estos datos, puede suponer un gran impacto en la mejora del proceso educativo, pero dada la gran cantidad de datos de bajo nivel que se recogen, analizarlos por parte de los usuarios puede resultar una tarea muy tediosa o prácticamente imposible en muchos casos.

La plataforma Open edX, creada por la Universidad de Harvard y el Instituto Tecnológico de Massachusetts (MIT), con la colaboración de múltiples universidades e instituciones de todo el mundo, es una de las plataformas MOOC con más éxito en el momento, contando con más de 400 cursos y millones de usuarios. Debido al temprano grado de desarrollo de la plataforma, aún no cuenta con un módulo completo que aplique técnicas de *Learning Analytics* para interpretar los datos que se recogen y mostrarlos de forma que los usuarios puedan interpretarlos fácilmente.

El objetivo de este proyecto es, aprovechando la facilidad de desarrollo en una plataforma de código abierto, realizar una primera aproximación al diseño de un módulo de tratamiento de datos, con objeto de ofrecerlo a la comunidad para su mejora y mantenimiento, pensando, en un futuro, en su posible integración con la plataforma Open edX, de forma que cualquier usuario pueda beneficiarse de él. En este trabajo se colabora en el diseño de la arquitectura de este módulo, incluyendo una capa intermedia para obtención de datos de la plataforma, de forma que pueda ser empleada para obtener datos tanto por las visualizaciones del módulo como por otros desarrolladores que quieran hacer uso de ella. También, en este trabajo, se diseñan e implementan una serie de visualizaciones para dicho módulo, incluyendo temas como notas de los alumnos, acceso a las secciones del curso, o progreso de los alumnos en vídeos y problemas.



ÍNDICE GENERAL

1.	INTRODUCCIÓN Y OBJETIVOS	1
1.1.	MOTIVACIÓN.....	1
1.2.	OBJETIVOS.....	3
1.2.1.	<i>Familiarización con la plataforma Open edX y las tecnologías empleadas para trabajar en ella</i>	3
1.2.2.	<i>Diseñar un “API” para la obtención de datos de la plataforma Open edX</i>	3
1.2.3.	<i>Desarrollar un conjunto de visualizaciones de analítica del aprendizaje para Open edX</i>	3
1.2.4.	<i>Propuesta e implementación de arquitectura para integrar el módulo de analítica del aprendizaje en Open edX</i>	4
1.3.	PLANIFICACIÓN	4
1.4.	ESTRUCTURA DE LA MEMORIA.....	6
2.	ESTADO DEL ARTE	8
2.1.	PLATAFORMAS MOOC.....	8
2.1.1.	<i>edX</i>	9
2.1.2.	<i>Alison</i>	9
2.1.3.	<i>Udacity</i>	10
2.1.4.	<i>Coursera</i>	10
2.1.5.	<i>Khan Academy</i>	10
2.1.6.	<i>Otras (MOOC en países hispanohablantes)</i>	11
2.2.	BIG DATA Y LEARNING ANALYTICS.....	11
2.2.1.	<i>Learning analytics</i>	13
	El Ciclo de Learning Analytics	
	Learning Analytics en Plataformas MOOC	
	Learning Analytics en EDX	
2.3.	TECNOLOGÍAS EMPLEADAS.....	20
2.3.1.	<i>Tecnologías empleadas en el desarrollo de Open edX</i>	21
2.3.2.	<i>Nuevas tecnologías introducidas para el desarrollo de un módulo de learning analytics</i>	22
	Scheduler: Celery beat	
	Visualizaciones: Google Charts	
3.	ARQUITECTURA.....	24
3.1.	ARQUITECTURA OPEN EDX.....	24
3.1.1.	<i>Arquitectura del LMS</i>	25
3.1.2.	<i>Courseware (Cursos)</i>	26
3.1.3.	<i>Gestión de datos: Bases de datos</i>	27
3.1.4.	<i>Gestión de datos: Tracking Logs</i>	28
3.2.	ARQUITECTURA INTERNA DEL MÓDULO DE LEARNING ANALYTICS.....	29
3.2.1.	<i>Arquitectura</i>	30
3.2.2.	<i>Modelo: Tablas en base de datos</i>	31
3.2.3.	<i>Vista: views.py, plantillas y static files</i>	33
3.2.4.	<i>Controlador: API de datos</i>	35
3.2.5.	<i>Controlador: Cálculo de estadísticas de alto nivel</i>	36
3.2.6.	<i>Controlador: Tareas periódicas</i>	38
4.	API DE DATOS.....	40
4.1.	CURSOS.....	41

4.2.	PROBLEMAS.....	43
4.3.	EVENTOS.....	44
4.4.	LOCALIZADORES, KEYS Y URLS.....	45
5.	VISUALIZACIONES	48
5.1.	COURSE SUMMARY	48
5.1.1.	<i>Motivaciones y utilidad</i>	48
5.1.2.	<i>Modo de empleo</i>	49
5.1.3.	<i>Funcionamiento interno</i>	49
5.2.	STUDENT GRADES	50
5.2.1.	<i>Motivaciones y utilidad</i>	51
5.2.2.	<i>Modo de empleo</i>	51
5.2.3.	<i>Funcionamiento interno</i>	52
5.3.	CHAPTER TIME.....	54
5.3.1.	<i>Motivaciones y utilidad</i>	54
5.3.2.	<i>Modo de empleo</i>	55
5.3.3.	<i>Funcionamiento interno</i>	55
5.4.	CHAPTER ACCESES.....	58
5.4.1.	<i>Motivaciones y utilidad</i>	58
5.4.2.	<i>Modo de empleo</i>	58
5.4.3.	<i>Funcionamiento interno</i>	59
5.5.	PROBLEM & VIDEO PROGRESS.....	61
5.5.1.	<i>Motivaciones y utilidad</i>	62
5.5.2.	<i>Modo de empleo</i>	62
5.5.3.	<i>Funcionamiento interno</i>	62
6.	CONCLUSIONES, RESULTADOS Y TRABAJOS FUTUROS.....	65
6.1.	CONCLUSIONES.....	65
6.1.1.	<i>Familiarización con la plataforma Open edX y las tecnologías empleadas para trabajar en ella</i>	65
6.1.2.	<i>Diseñar un “API” para la obtención de datos de la plataforma Open edX</i>	66
6.1.3.	<i>Desarrollar un conjunto de visualizaciones de analítica del aprendizaje para Open edX</i>	66
6.1.4.	<i>Propuesta e implementación de arquitectura para integrar el modulo de analítica del aprendizaje en Open edX</i>	67
6.2.	PRUEBAS Y RESULTADOS	67
6.2.1.	<i>Evaluación de ANALYSE</i>	67
6.2.2.	<i>Publicaciones presentadas a la comunidad</i>	68
6.3.	TRABAJOS FUTUROS.....	69
6.3.1.	<i>Escalabilidad</i>	69
6.3.2.	<i>Nuevas visualizaciones</i>	70
6.3.3.	<i>Mantenimiento y soporte de bugs</i>	70
6.3.4.	<i>Mejora de integración y sencillez de instalación</i>	71
7.	REFERENCIAS.....	72

ÍNDICE DE FIGURAS

FIGURA 1: DIAGRAMA DE GANTT DE LA PLANIFICACIÓN DEL WBS (WORK BREAKDOWN STRUCTURE)	5
FIGURA 2: DIFERENCIA ENTRE LEARNING ANALYTICS Y ACADEMIC ANALYTICS [19]	12
FIGURA 3: CICLO DE LEARNING ANALYTICS, POR CLOW, 2012	15
FIGURA 4: CALIFICACIONES DE TODAS LAS SECCIONES EVALUADAS	17
FIGURA 5: DESGLOSE DE LAS CALIFICACIONES DE CADA CAPÍTULO DEL CURSO	17
FIGURA 6: ANALYTICS - ESTADÍSTICAS DEL CURSO	18
FIGURA 7: DATA DOWNLOAD DEL DASHBOARD DE INSTRUCTOR	18
FIGURA 8: CAPTURA DE UNA DE LAS VISUALIZACIONES DE EDX INSIGHTS	20
FIGURA 9: PATRÓN MODELO-VISTA-CONTROLADOR	21
FIGURA 10: ARQUITECTURA GENERAL DE OPEN EDX	25
FIGURA 11: ESTRUCTURA DE LOS CURSOS EN OPEN EDX	26
FIGURA 12: BASES DE DATOS EN LA ARQUITECTURA DE OPEN EDX	28
FIGURA 13: MÓDULO DE LEARNING ANALYTICS EN LA ARQUITECTURA DE OPEN EDX	30
FIGURA 14: DIVISIONES DE LAS VISUALIZACIONES	34
FIGURA 15: VISUALIZACIÓN «COURSE SUMMARY»	35
FIGURA 16: VISUALIZACIÓN “COURSE SUMMARY”	48
FIGURA 17: VISUALIZACIÓN “STUDENT GRADES”	51
FIGURA 18: VISUALIZACIÓN “CHAPTER TIMES”	54
FIGURA 19: INTERACCIÓN CON LA VISUALIZACIÓN “CHAPTER TIMES”	55
FIGURA 20: DIAGRAMA DE FLUJO DE LAS FUNCIONES <code>MANAGE_BROWSER_EVENT</code> Y <code>MANAGE_SERVER_EVENT</code>	56
FIGURA 21: VISUALIZACIÓN “CHAPTER ACCESES”	58
FIGURA 22: INTERACCIÓN CON LA VISUALIZACIÓN “CHAPTER ACCESES” (SEQUENTIALS)	59
FIGURA 23: INTERACCIÓN CON LA VISUALIZACIÓN “CHAPTER ACCESES” (VERTICALS)	59
FIGURA 24: VISUALIZACIÓN “PROBLEM & VIDEO PROGRESS”	61

Índice DE TABLAS

TABLA 1 TAREAS Y TEMPORALIZACIONES	5
TABLA 2: COURSESTRUCT - CAMPOS DE LA TABLA DE LA BASE DE DATOS	32
TABLA 3: STUDENTGRADES - CAMPOS DE LA TABLA DE LA BASE DE DATOS	32
TABLA 4: FUNCIÓN UPDATE_DB_COURSE_STRUCT	36
TABLA 5: FUNCIÓN GET_DB_COURSE_STRUCT	37
TABLA 6: FUNCIÓN DEL API GET_COURSES_LIST	41
TABLA 7: FUNCIÓN DEL API GET_COURSE_KEY	41
TABLA 8: FUNCIÓN DEL API GET_COURSE_MODULE	41
TABLA 9: FUNCIÓN DEL API GET_COURSE_STRUCT	41
TABLA 10: FUNCIÓN DEL API GET_COURSE_GRADE_CUTOFF	42
TABLA 11: FUNCIÓN DEL API GET_COURSE_STUDENTS	42
TABLA 12: FUNCIÓN DEL API DUMP_FULL_GRADING_CONTEXT	42
TABLA 13: FUNCIÓN DEL API GET_COURSE_BLOCKS	43
TABLA 14: FUNCIÓN DEL API IS_PROBLEM_DONE	43
TABLA 15: FUNCIÓN DEL API SECTION_MAX_GRADE	43
TABLA 16: FUNCIÓN DEL API GET_PROBLEM_SCORE	44
TABLA 17: FUNCIÓN DEL API IS_PROBLEM_FOR_COURSE	44
TABLA 18: FUNCIÓN DEL API GET_COURSE_EVENTS_SQL	44
TABLA 19: FUNCIÓN DEL API GET_COURSE_ACCESS_EVENTS_SQL	45
TABLA 20: FUNCIÓN DEL API GET_PROBLEM_HISTORY_SQL	45
TABLA 21: FUNCIÓN DEL API IS_SAME_COURSE	46
TABLA 22: FUNCIÓN DEL API GET_COURSE_FROM_URL	46
TABLA 23: FUNCIÓN DEL API GET_LOCATIONS_FROM_URL	46
TABLA 24: FUNCIÓN DEL API COMPARE_LOCATIONS	46
TABLA 25: TABLA DE LA BASE DE DATOS SORTGRADES	50
TABLA 26: TABLA DE LA BASE DE DATOS STUDENTGRADES	52
TABLA 27: TABLA DE LA BASE DE DATOS COURSETIME	57
TABLA 28: TABLA DE LA BASE DE DATOS COURSEACCESSES	61
TABLA 29: TABLA DE LA BASE DE DATOS COURSEPROBVIDPROGRESS	64

CAPÍTULO 1

1. INTRODUCCIÓN Y OBJETIVOS

En este primer capítulo se explica cual es la motivación para la realización del proyecto. También se presenta una breve descripción de los objetivos a cumplir en el proyecto. Por último, se incluye una planificación del trabajo a realizar, así como una descripción del contenido de cada uno de los capítulos en los que esta dividida esta memoria.

1.1. MOTIVACIÓN

Los cursos on-line masivos, o MOOC (Massive Online Open Course), se han popularizado mucho en los últimos años como plataformas para la educación a distancia. Algunas de las características de los MOOC que han motivado el crecimiento de estos cursos son: el carácter *open source* de estos, contenidos alojados on-line y cursos enfocados a llegar a un gran número de alumnos que acceden de forma masiva y gratuita.

La educación abierta que proporcionan los MOOC a través de la red, permite la participación interactiva a gran escala de cientos de personas que tienen la oportunidad de acceder a una gran variedad de cursos de calidad, independientemente de su localización o nivel económico. Sumando el hecho de que muchas de las universidades más importantes del mundo hayan empezado a usar plataformas MOOCs para difundir sus cursos, se está permitiendo el acceso a educación de alto nivel a personas que por una razón u otra, no podrían permitírselo con las plataformas educativas tradicionales.

Al mismo tiempo, el trabajo de los docentes también se facilita mucho al emplear estos cursos, dado que con un trabajo similar al necesario para un curso tradicional, pueden llegar a miles de alumnos en todo el mundo, y pueden aprovechar muchas de las herramientas sociales de estos MOOCs, para que los propios alumnos se encarguen de evaluarse y ayudarse entre ellos.

Una de las características más importantes de estos cursos, es la gran cantidad de alumnos que los componen. Debido a este alto número de usuarios, los cursos MOOC pueden recoger una gran cantidad de datos generados por las iteraciones de estos usuarios con el curso. Esto puede suponer una gran ventaja, ya que es posible obtener estadísticas muy fiables para mejorar el proceso educativo, debido a la gran cantidad de muestras recogidas. Pero también puede suponer una desventaja si la plataforma que ofrece los cursos no cuenta con una buena herramienta para tratar y presentar de forma sencilla estos datos, ya que debido al gran volumen de estos, puede resultar muy complicado analizarlos por parte de los instructores y usuarios.

Entre las plataformas agregadoras de cursos MOOC se pueden encontrar algunas muy extendidas como [Udacity](#), [Coursera](#) o [Khan Academy](#). También existen otras más nuevas, o menos extendidas, como es el caso de [Miríada X](#), desarrollada a través de la Red Universia, centrada en universidades e instituciones iberoamericanas, y en la que colaboran multitud de universidades españolas. Dentro de estas nuevas plataformas MOOC, hay una que destaca especialmente por el rápido crecimiento que ha tenido desde su aparición: [Open edX](#), fundada por el Instituto Tecnológico de Massachussetts y la Universidad de Harvard, y con la que colaboran multitud de universidades de todo el mundo, así como importantes instituciones tales como la Linux Foundation o el Fondo Monetario Internacional[1]. edX, a pesar de haber sido lanzada más tarde que otras plataformas como Udacity o Coursera, ofrece cursos de las mejores universidades del mundo [1] y esta igualando a estas en número de usuarios y cursos, por lo que puede ser considerada como una de las plataformas MOOC más importantes en la actualidad.

La plataforma Open edX, a pesar de tener un potencial altísimo, aún tiene ciertas carencias, debido principalmente a su temprano desarrollo. Entre estas carencias, una de las que más destacan es la falta de un módulo complejo de análisis y presentación de los datos recogidos en los cursos. Como se ha comentado anteriormente, debido al carácter masivo de los MOOCs, cada curso proporciona una cantidad de datos muy elevada y por lo tanto, muy difícil de analizar por parte de los docentes, por lo que es necesario, o al menos muy positivo para el proceso educativo, contar con una herramienta capaz de tratar, simplificar, y presentar de manera sencilla estos datos. De esta forma, gracias a estas estadísticas de alto nivel, los docentes podrán tener una idea general del curso rápidamente, y los alumnos podrán tener una mejor idea sobre su progreso en el curso. La plataforma edX ya ofrece algunas visualizaciones y estadísticas, tanto al alumno como al docente. Pero estas estadísticas y visualizaciones, se quedan cortas a la hora de proporcionar un análisis detallado del curso que pueda servir al instructor para mejorar el curso, y realizar un seguimiento completo de la evolución general de los alumnos.

Por tanto, la motivación de este proyecto surge a la vista del éxito que está teniendo edX en el campo de la educación online, el hecho de que el código de la plataforma este abierto a la comunidad, y sobre todo el impacto positivo que tendría en la plataforma la existencia de un módulo que presente estadísticas de alto nivel a los instructores y alumnos.

Este proyecto, contribuirá, junto a otros proyectos y tesis realizadas en paralelo en la Universidad Carlos III de Madrid, a diseñar e implementar un módulo para la plataforma Open edX encargado de obtener, tratar y presentar a los usuarios diversos datos de la plataforma, de forma que sea posible analizar y mejorar el proceso educativo por parte de los instructores, así como consultar el progreso en el curso por parte de los alumnos. Para ello se diseñará por una parte la arquitectura del módulo, incluyendo un API general de obtención de datos de la plataforma, que pueda ser aprovechada por parte de la comunidad en otros desarrollos. También se diseñarán e implementarán una serie de visualizaciones que traten estos datos y los muestren de forma sencilla a los usuarios.

1.2. OBJETIVOS

A continuación se comentarán los objetivos planeados para la realización de este proyecto.

1.2.1. Familiarización con la plataforma Open edX y las tecnologías empleadas para trabajar en ella

Dado que Open edX es una plataforma abierta con un nivel de complejidad bastante alto, el primer objetivo de este proyecto será conocer a fondo la arquitectura de este desarrollo, así como las tecnologías empleadas para este.

El primer paso será analizar a fondo la arquitectura de la plataforma: cual es la estructura del código, que tipos de bases de datos se emplean y familiarizarse con sus tablas, así como estudiar los desarrollos anteriores que pueden ser útiles para el modulo que se va a diseñar.

Una vez estudiada esta arquitectura, será importante conocer las tecnologías que se emplean para desarrollar en la plataforma, tales como el framework de desarrollo que emplean (Django), lenguajes (Python, JavaScript, Html), bases de datos (MySQL, MongoDB), etc.

Por último, debido al gran número de colaboradores es necesario saber cual es la forma correcta de colaboración dentro de su proyecto, alojado en GitHub.

1.2.2. Diseñar un “API” para la obtención de datos de la plataforma Open edX

El proyecto Open edX cuenta actualmente, con varias herramientas para acceder a los datos almacenados en las diversas bases de datos (Xmodule, Django con MySQL, etc), pero no cuenta con ninguna capa intermedia específicamente definida para obtener datos y estadísticas de alto nivel que puedan ser útiles para un módulo similar al que se plantea diseñar. Por tanto, el primer objetivo de este proyecto, será diseñar esta capa intermedia, que trate los datos de bajo nivel que nos ofrece la plataforma con el fin de poder usarlos en el módulo de *learning analytics* que se va a implementar.

Además, dado que Open edX es un proyecto abierto, esta API también puede ser de mucha utilidad a personas que decidan ampliar este módulo en un futuro, o utilizarlo para desarrollos propios.

1.2.3. Desarrollar un conjunto de visualizaciones de analítica del aprendizaje para Open edX

Como se comenta en el apartado anterior, las herramientas con las que cuenta edX para el tratamiento de datos, y el seguimiento de los cursos, aunque existentes, son muy escasas. Por tanto, el grueso de este proyecto, se centrará en el desarrollo de un módulo con estadísticas y visualizaciones de alto nivel, haciendo uso del API que se comenta en el punto anterior y de

utilidad para los docentes que podrán obtener un *feedback* complejo del estado del curso: carencias, puntos fuertes y puntos débiles de éste, así como detalles de dificultad, utilidad, o interés por parte del alumno en elementos concretos del curso, tales como problemas o vídeos. Los alumnos también tendrán sus propias visualizaciones que les ayudarán a tener un conocimiento más concreto sobre su proceso de aprendizaje y su progreso en el curso.

En concreto, en este proyecto, se diseñarán e implementarán 5 de las visualizaciones que formaran parte del módulo, mostrando datos sobre las calificaciones de los alumnos, los accesos y tiempos dedicados a cada una de las secciones del curso y el progreso de los alumnos en los vídeos y problemas del curso.

1.2.4. Propuesta e implementación de arquitectura para integrar el módulo de analítica del aprendizaje en Open edX

Una vez extraídos los datos de la plataforma, y realizadas las estadísticas y visualizaciones, el siguiente paso será integrar estas visualizaciones en el LMS (Learning Management System) de forma que puedan ser consultadas fácilmente por instructores y alumnos, diseñando una arquitectura para el módulo que la integre dentro de la plataforma Open edX de forma que sea consecuente con el diseño de la arquitectura del resto de la plataforma.

Se añadirán varias opciones, especialmente para instructores, como por ejemplo seleccionar estadísticas por grupos de alumnos, o seleccionar vídeos o problemas concretos.

Otra parte importante de la integración serán la seguridad y la eficiencia del módulo. Por una parte será necesario diseñar una forma de tratar los datos sin sobrecargar el servidor sobre el que se aloje la plataforma, o el cliente que cargue las visualizaciones. Por otra parte habrá consideraciones de seguridad, como que los usuarios solo puedan acceder a los datos de los cursos en los que están inscritos, e incluso dentro de un mismo curso, que los alumnos no puedan acceder a datos de otros alumnos, o a datos que solo el instructor debería tener acceso.

1.3. PLANIFICACIÓN

En este apartado se muestra la planificación para la realización del proyecto. En la *Tabla 1* se pueden observar las tareas junto con la duración de estas. También se indican los predecesores de ciertas tareas, es decir las tareas que han debido ser completadas antes de poder comenzar con la tarea indicada. En la *Figura 1* se muestra el diagrama de Gantt de la planificación, indicando el WBS (Work Breakdown Structure) de cada tarea. Nótese que la duración de las tareas se expresa en horas y días, y que la relación entre estos variara según la fecha, esto es debido a diversas razones, que permiten más o menos horas por día según el periodo de tiempo en que se realicen.

Tabla 1 Tareas y temporalizaciones

WBS	Tarea	Comienzo	Fin	Duración	Predecesores
1	Familiarización con el LMS, CMS y cursos de la plataforma edX	4/11/2013	10/11/2013	15h (7 días)	
2	Estudio de tecnologías empleadas en edX: Python, Django Web Framework, JavaScript, Mako y CSS	11/11/2013	22/12/2013	90h (42 días)	
3	Periodo de descanso por Navidad	23/12/2013	5/1/2014	14 días	
4	Estudio de la instalación y metodología de desarrollo de la plataforma Open edX	6/1/2014	12/1/2014	15h (7 días)	
5	Estudio del código y arquitectura de la plataforma Open edX	13/1/2014	23/2/2014	90h (42 días)	1,2,4
6	Definición de visualizaciones para el módulo y estudio de datos necesarios	24/2/2014	2/3/2014	15h (7 días)	5
7	Diseño de la arquitectura del módulo y creación del proyecto y configuración dentro de la plataforma Open edX	3/3/2014	23/3/2014	45h (21 días)	5
8	Diseño y codificación del API de obtención de datos	24/3/2014	3/8/2014	150h (133 días)	6,7
9	Descanso por vacaciones de verano	4/8/2014	17/8/2014	14 días	
10	Diseño y codificación de las visualizaciones	18/8/2014	21/12/2014	150h (126 días)	6,7
11	Descanso por vacaciones de navidad	22/12/2014	4/1/2015	14 días	
12	Integración de las visualizaciones y el módulo en la plataforma Open edX	5/1/2015	5/4/2015	150h (91 días)	10
13	Pruebas del módulo y arreglo de bugs encontrados	6/4/2015	14/6/2015	150h (70 días)	12
14	Diseño y redacción de la memoria del PFC	15/6/2015	9/8/2015	120h (56 días)	

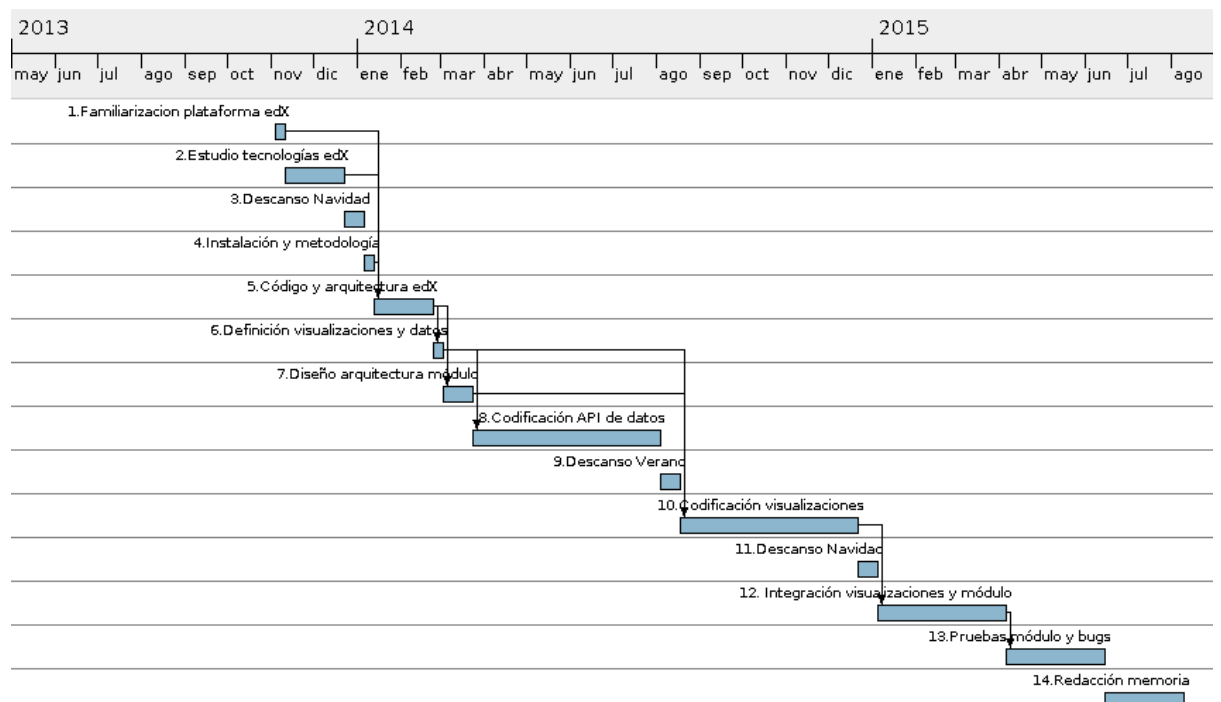


Figura 1: Diagrama de Gantt de la planificación del WBS (Work Breakdown Structure)

1.4. ESTRUCTURA DE LA MEMORIA

En este apartado se comenta brevemente el contenido de cada uno de los capítulos y apéndices de la memoria.

Capítulo 1: Introducción

Este primer capítulo es una introducción a la memoria en el que se comentan las motivaciones existentes para la realización de este proyecto, así como los objetivos planeados para afrontarlo. También se incluirá una planificación del tiempo que se dedicará a cada uno de los objetivos.

Capítulo 2: Estado del arte

En este capítulo se incluye un análisis sobre los conceptos y tecnologías que rodean al proyecto y que puedan ser de utilidad para el desarrollo de este.

Se realiza un análisis sobre los cursos MOOC y las plataformas existentes, así como un estudio sobre los conceptos *big data* y *learning analytics* y su presencia en plataformas MOOC. Posteriormente se analizará el estado concreto de Open edX, especialmente en la aplicación de técnicas de *learning analytics* en la plataforma. Por último, se realiza un análisis de las tecnologías que se emplean en Open edX y las nuevas que se van a emplear para el desarrollo del proyecto.

Esta sección es de especial importancia debido al temprano estado de desarrollo en el que se encuentra el proyecto Open edX, especialmente respecto a la aplicación de técnicas de *learning analytics* dentro de la plataforma.

Capítulo 3: Arquitectura

En el tercer capítulo se describe la arquitectura del módulo de *learning analytics* que se va a implementar, así como la relación con el resto de la arquitectura de Open edX.

Primero se analiza la arquitectura de la plataforma Open edX, centrándose en las partes de esta que más importancia tienen para la aplicación en el módulo de *learning analytics*. A

continuación se presentará un diseño de arquitectura para dicho módulo y cada una de sus partes, así como su lugar dentro de la arquitectura de la plataforma. Por último se presentan en detalle algunas de las partes más importantes del módulo, como el encolador de tareas que se encarga de realizar las estadísticas en segundo plano, o los modelos Django encargados de almacenar las estadísticas calculadas en las bases de datos.

Capítulo 4: API de datos

En este capítulo se presenta el API creado como capa intermedia para la obtención de datos de la plataforma Open edX por parte del módulo de *learning analytics* que se desarrolla en este proyecto. Además, este API también puede ser de utilidad para otros desarrolladores de la plataforma, que trabajen en el desarrollo de este módulo o en desarrollos propios.

Se especificarán las motivaciones para la creación de este modulo así como los criterios de diseño de las funciones que se incluyen en el. En concreto, las funciones que se incluyen en el API por parte de este proyecto, se incluirán en 4 grupos, según los datos que ofrezcan: cursos, problemas, eventos, y localizadores.

Capítulo 5: Visualizaciones

En este capítulo se hace una descripción de cada una de las visualizaciones que se incluyen en el módulo de *learning analytics*.

Se incluyen detalles de implementación del cálculo de la estadística, usabilidad y posibilidades de interacción con la visualización, así como un análisis de la utilidad de la visualización, tanto para el instructor como para los alumnos.

Capítulo 6: Conclusiones, resultados y trabajos futuros

En este último capítulo de la memoria, se analiza, por una parte, si se han cumplido satisfactoriamente los objetivos planteados al inicio del proyecto. Por otra parte, se presentan los resultados que han ofrecido el desarrollo de este proyecto, tales como evaluaciones del módulo o artículos presentados a la comunidad sobre dicho módulo. Por último, se analizan cuales pueden ser los siguientes pasos a dar en el desarrollo del módulo diseñado.

CAPÍTULO 2

2. ESTADO DEL ARTE

En este capítulo se hace un análisis general del estado de los distintos conceptos relacionados con este proyecto.

En primer lugar, se realiza un estudio de las principales plataformas MOOC que existen en la actualidad, analizando sus particularidades, objetivos y estado actual.

Posteriormente se analizan conceptos como *Big Data* o *Learning Analytics* y las ventajas de la aplicación de estas técnicas a la educación online, tanto las que ya se emplean, como las posibles aplicaciones futuras.

Una vez introducidos estos conceptos se estudia la aplicación que tienen actualmente estas técnicas en la plataforma Open edX, así como las posibles aplicaciones que podría tener el desarrollo de este proyecto en el futuro de la plataforma.

Por ultimo se hace un estudio de las principales tecnologías que se emplean para el desarrollo en la plataforma Open edX, así como nuevas tecnologías y herramientas que se emplearán para alcanzar el objetivo del proyecto.

2.1. PLATAFORMAS MOOC

Los MOOC están revolucionando las formas clásicas de la educación online, ofreciendo cursos de instituciones de primer nivel, abiertos al mundo entero, y en los que pueden participar un gran número de alumnos. En los últimos años estas plataformas están creciendo en numero de usuarios muy rápidamente[2]sustituyendo a las plataformas tradicionales de educación online, basadas en LMS (Learning Management Systems)

Los MOOC, a pesar de tener muchas similitudes con las plataformas LMS tradicionales, se diferencian de ellas en varios aspectos. La diferencia más notable se encuentra en el número de alumnos que participan en dichos cursos, mientras las plataformas de educación on-line tradicionales suelen ofrecer cursos para un número de usuarios similar al de una clase presencial, el objetivo de los MOOC es ofrecer cursos para miles de alumnos. A raíz de esta diferencia, nos encontramos con que el planteamiento de los MOOC difiere en gran medida de las plataformas tradicionales, mientras estas ultimas están pensadas para asistir a una clase ya existente de forma presencial, los MOOC pretenden sustituir esa clase por la plataforma on-line. Debido a esto, el perfil de los alumnos y los instructores será distinto al que nos encontramos en las plataformas tradicionales. El perfil de los alumnos de cursos MOOC cambia drásticamente en relación con el del alumno de un curso tradicional, encontrándonos

con un gran número de alumnos que, o bien se inscribe en el curso sin una intención firme de realizarlo, o no es capaz de organizarse correctamente para la educación a distancia. Esto se traduce en que en los cursos MOOC, la tasa de aprobados es muy baja, en torno al 5-10% de los alumnos inscritos. Por tanto, el perfil del profesor también tiene que ser distinto al de un curso tradicional, necesitando personas que además de ser buenos educadores, tengan habilidades de diseño instruccional, diseño de cursos y monitorización de estudiantes. [3]

A pesar de la reciente aparición del concepto, ya existen múltiples plataformas pensadas para ofrecer esta clase de cursos. La plataforma edX, es una de las más importantes de la actualidad, con un rápido crecimiento en los últimos años. Pero edX, no es ni la primera ni la única plataforma MOOC, a continuación, se hace un repaso de las más importantes en la actualidad, analizando su difusión, objetivos, así como las características que las diferencian del resto de plataformas MOOC.

2.1.1. edX

Es una plataforma fundada en 2012 por la Universidad de Harvard y el Instituto Tecnológico de Massachusetts (MIT).

El objetivo principal es ofrecer cursos de nivel universitario abiertos a todo el mundo y de forma gratuita (aunque para obtener algunos certificados es necesario el pago para el mantenimiento de la plataforma [4]). Estos cursos se basan en bloques de aprendizaje semanales, compuestos por vídeos, textos, así como todo tipo de ejercicios para evaluar los conocimientos adquiridos. La plataforma también está enfocada a la colaboración activa entre los usuarios, por lo que los foros de discusión de las lecciones, son una importante parte de los cursos, en las que tanto alumnos como profesores participan para aclarar conceptos, o discutir sobre temas tratados en la lección. Por último, otro de los objetivos de la plataforma y de las entidades que participan en ella, es obtener datos del uso de los cursos para investigar cómo aprenden los alumnos y cómo la tecnología puede revolucionar el aprendizaje, con el objetivo de aplicar las conclusiones en sus propios campus [5].

Actualmente, más de 60 universidades, entidades sin ánimo de lucro, grandes empresas y organismos internacionales, colaboran con edX. Dentro de estas se encuentran muchas de las más importantes universidades del mundo, y entidades tan importantes como Microsoft o la Linux Foundation. En la actualidad es una de las plataformas MOOC con más éxito del mundo, teniendo un rápido crecimiento desde el momento de su lanzamiento, contando en la actualidad con más de 400 cursos y más de 3 millones de usuarios registrados. [1] [6]

EdX es un proyecto sin ánimo de lucro, y la plataforma se ha desarrollado como software open-source, abriendo el código de esta a otras instituciones en Junio de 2013 y recibiendo el nombre de Open edX [7]. Universidades de todo el mundo, y grandes empresas como Google, colaboran activamente en el desarrollo de la plataforma.

2.1.2. Alison

Es considerada por muchos como la primera plataforma en alojar cursos MOOC. ALISON es una plataforma, fundada en Galway (Irlanda) en 2007, cuyo objetivo es ofrecer cursos certificados, tanto para educación básica como para formación para el entorno profesional.

Aunque los usuarios pueden registrarse en la plataforma desde prácticamente cualquier parte del mundo, muchos de sus estudiantes provienen de países en vías de desarrollo, a diferencia de otras plataformas MOOC, cuyo grueso de estudiantes proviene de Estados Unidos. Es una de las plataformas MOOC más grandes fuera de Estados Unidos, con más de 3 millones de usuarios registrados en Febrero de 2014. [8]

ALISON es una plataforma comercial, pero ofrece todos sus cursos de manera gratuita. Algunas de las entidades que ofrecen cursos en la plataforma, lo hacen de manera gratuita, otras se financian mediante certificados de pago, o publicidad PPC (Pay-Per-Click) [9]. De esta manera se consigue, que los usuarios de los países desarrollados paguen, en cierta medida, la educación de aquellos en países en vías de desarrollo.

2.1.3. Udacity

Es una plataforma fundada por académicos de la Universidad de Stanford y lanzada en febrero de 2012. Udacity surge a raíz de unas clases online gratuitas sobre computación ofrecidas en 2011 por Sebastian Thrun a través de la Universidad de Stanford. A la vista del éxito de las clases (más de 160.000 alumnos), se decidió fundar Udacity, con la ayuda de una entidad de capital riesgo. [10]

Aunque en un principio la plataforma ofrecía cursos universitarios, ofrecidos por varias universidades, en los últimos años se ha centrado más en ofrecer cursos vocacionales y de formación para el entorno profesional. [11]

2.1.4. Coursera

Coursera es una plataforma lanzada en abril de 2012, también por académicos de la Universidad de Stanford. Es una de las plataformas MOOC más grandes, con más de 10 millones de usuarios y más de 500 cursos [12]. Su objetivo es ofrecer cursos gratuitos de nivel universitario, abiertos a todos los sectores de la población mundial.

Al contrario que Udacity y otras plataformas MOOC, cuyo forma de evaluación era principalmente por medio de ejercicios corregidos automáticamente o uso de *robo-graders*, haciendo difícil la oferta de cursos de temática no técnica o científica, el objetivo de Coursera era ofrecer también cursos relacionados con temáticas como las artes o las ciencias sociales. Para poder evaluar este tipo de cursos, los desarrolladores emplean técnicas de peer-review y *feedback* entre alumnos. De esta forma los alumnos pueden evaluarse y ayudarse unos a otros, de acuerdo a las indicaciones de los profesores. [13]

Coursera es una plataforma comercial, y emplea todo tipo de maneras para financiar los cursos, como cobros por certificados, presentar estudiantes a empresarios potenciales, tutorías, matriculas, etc.

2.1.5. Khan Academy

Khan Academy es una organización de e-learning sin ánimo de lucro creada en 2006 por Salman Khan, cuyo objetivo es "proporcionar una educación de nivel mundial para cualquier persona, en cualquier lugar. [14]

A pesar de que ofrece educación online gratuita de forma masiva, esta plataforma no se engloba dentro de la definición de MOOC, ya que esta organizada en forma de lecciones cuyo centro son vídeos alojados en youtube, en lugar de en forma de curso tradicional. [15]

A pesar de esto, Khan Academy es una de las plataformas con un mayor desarrollo de módulos de estadísticas y *learning analytics*, y dado que muchos de estos conceptos son igual de aplicables en esta plataforma que en un MOOC tradicional, se estudiarán alguno de estos módulos para diseñar las visualizaciones del modulo de *learning analytics* para Open edX, como por ejemplo, el módulo ALAS-KA [16]desarrollado en la Universidad Carlos III de Madrid.

Como se ha comentado, la plataforma no tiene ánimo de lucro, y tampoco se financia mediante publicidad o suscripciones, si no que se mantiene a base de donaciones y el trabajo de los voluntarios. [14]

2.1.6. Otras (MOOC en países hispanohablantes)

A pesar de que la mayoría de las plataformas MOOC tienen su origen en Estados Unidos, el fenómeno MOOC también ha llegado a los países hispanohablantes en los últimos años. Las dos plataformas más importantes en este ámbito son Miríada X y Wedubox.

Miríada X surge en 2013 como un proyecto del Banco Santander y Telefónica, a través de la Red Universia y Telefónica Educación Digital basándose en la plataforma de software libre WEMOOC. Actualmente es la plataforma MOOC más grande de Iberoamérica y una de las más grandes a nivel mundial[17]. Su sistema docente se basa en impartir cursos creados por múltiples Universidades de España y Latinoamérica de forma gratuita y dirigidos a cualquier usuario interesado, independientemente de su nivel académico.

Por otra parte, Wedubox nace en septiembre de 2012, como una red de educación fundamentada en el *crowdsourcing* social, es decir, un lugar donde miles de docentes y expertos pueden crear cursos basados en su conocimiento, y ponerlos a disposición de cualquier persona interesada. En el último año, los creadores de Wedubox, acuñaron el termino E2N (Ecosistema de Educación en la Nube) como la nueva generación de LMS e integraron sus principios en la nueva generación de Wedubox, Eduvolucion. [18]

2.2. BIG DATA Y LEARNING ANALYTICS

En los últimos años el uso de plataformas online para todo tipo de gestiones (compras, negocios, comunicación, educación, etc) ha crecido exponencialmente, y por tanto también la cantidad de datos que estas plataformas registran.

El uso de estos datos para el beneficio de las compañías se ha convertido en una poderosa herramienta para diseñar nuevas técnicas de marketing, análisis de mercados y diseño de nuevos modelos de negocio. Términos como *data mining* o *business intelligence* han tomado gran relevancia en los últimos años, y muchas empresas están invirtiendo en estos análisis de grandes cantidades de datos para diseñar sus estrategias de negocio.

La aplicación de estas técnicas al campo de la educación online recibe el nombre de *Educational Data Mining* (EDM). Tratando las grandes cantidades de datos que se generan en las plataformas de e-learning, se pueden obtener diversos resultados que sirvan tanto a instructores o alumnos, como a plataformas educativas o universidades para mejorar cursos concretos, localizar puntos de abandono, alumnos en riesgo, etc, o a nivel más general, probar nuevos paradigmas de aprendizaje o realizar estudios de diversa índole.

Más recientemente algunos autores, como George Siemens, concluyeron que las técnicas de *Educational Data Mining* pueden dividirse en dos tipos: *Academic Analytics* y *Learning Analytics* (Figura 2). [19]

El primer termino, *Academic Analytics*, se refiere al conjunto de técnicas de análisis empleadas para mejorar los procesos de organización, la asignación de recursos, y en general cualquier proceso relacionado con mejorar la efectividad de una organización. Estas técnicas están enfocadas para ser aprovechadas por administradores, gobiernos, organismos de financiación y entidades internacionales, para mejorar y estudiar los procesos de aprendizaje de sus respectivas instituciones. [19]

TYPE OF ANALYTICS	LEVEL OR OBJECT OF ANALYSIS	WHO BENEFITS?
Learning Analytics	Course-level: social networks, conceptual development, discourse analysis, "intelligent curriculum"	Learners, faculty
	Departmental: predictive modeling, patterns of success/failure	Learners, faculty
Academic Analytics	Institutional: learner profiles, performance of academics, knowledge flow	Administrators, funders, marketing
	Regional (state/provincial): comparisons between systems	Funders, administrators
	National and International	National governments, education authorities

Figura 2: Diferencia entre *Learning Analytics* y *Academic Analytics* [19]

Por otra parte, el termino *Learning Analytics*, engloba las técnicas empleadas para realizar medidas y análisis sobre los estudiantes y su entorno, con el objetivo de entender y optimizar los procesos de aprendizaje en los que se encuentran, para mejorar el éxito de los alumnos. [20]

2.2.1. Learning analytics

Al igual que las técnicas de *business intelligence* sirven a las empresas para mejorar sus estrategias comerciales, las técnicas de *learning analytics* pueden ser muy útiles en el proceso educativo para todas las partes involucradas, entre otras cosas permiten:

- Predecir y encontrar alumnos que puedan necesitar apoyo extra
- Personalizar y adaptar el proceso de aprendizaje y los contenidos, asegurando que cada alumno reciba los recursos y forma de enseñanza acordes a su nivel de conocimiento.
- Motivar al estudiante proporcionándole información sobre su rendimiento y el de sus compañeros, así como sugiriéndole actividades y contenidos enfocados a reforzar conceptos concretos.
- Hacer mejor uso del tiempo y esfuerzo de los instructores proporcionando información sobre que alumnos necesitan ayuda extra, cuales pueden ser candidatos para ayudar a sus compañeros, o que métodos de enseñanza están ofreciendo mejores resultados.
- La visualización interactiva de información compleja ofrecerá a los instructores la capacidad de una visión concreta, o una visión global, dentro de conjuntos grandes de datos, dependiendo de las necesidades. [20]

La aplicación de estas técnicas es aún más importante en las plataformas MOOC, ya que debido a la gran cantidad de datos que estos generan, es muy difícil analizarlos sin emplear técnicas de *learning analytics*, perdiéndose una oportunidad tanto para atender mejor a los alumnos, como para estudiar y mejorar el proceso educativo en un futuro.

El Ciclo de Learning Analytics

En 2007, Campbell y Oblingerde, publican uno de los primeros artículos tratando el ciclo de aplicación de técnicas de *Educational Data Mining*[21]. Estos autores definen el ciclo de aplicación de técnicas analíticas en educación (en esta publicación se refieren a estas como *Academic Analytics*) como un proceso de 5 pasos:

- **Captura:**
Engloba el proceso de recogida de datos de los procesos educativos, incluyendo procesos de selección y organización de datos, políticas de administración y privacidad, etc.
- **Informes:**
Una vez capturados y almacenados los datos, el siguiente paso es presentarlos en forma de informe. Empleando herramientas de búsqueda, análisis y presentación, se intentará descubrir tendencias, patrones y excepciones en los datos. La presentación de los informes en forma de visualizaciones gráficas puede ser de gran utilidad, haciendo los informes más fáciles de interpretar.

- **Predicción:**

Analizando los informes de datos con diversas técnicas y modelos estadísticos, se obtienen predicciones sobre el proceso educativo tanto de forma general como de forma individual (secciones concretas, alumnos en riesgo, etc).

- **Actuar:**

Es el objetivo del proceso de *Academic Analytics*, una vez obtenidas las predicciones, actuar en consecuencia para mejorar el proceso educativo allá donde sea posible. Estas actuaciones pueden ser decididas entre los académicos al analizar las predicciones, o pueden realizarse de forma automática (por ejemplo, asignar refuerzo a alumnos en riesgo de abandonar el curso).

- **Refinar:**

Por último, los procesos analíticos deberían incluir métodos de mejora para los 4 pasos anteriores, observando el impacto de las actuaciones y mejorando los procesos estadísticos.

A pesar de que este análisis es una importante aproximación al proceso de *EDM*, esta pensado de forma genérica para todo el proceso, englobando tanto procesos de *learning analytics* como de *academic analytics*.

En un artículo posterior [22] Doug Clow, toma el modelo de Campbell y Obligerde, junto con otras teorías, para presentar el ciclo del proceso de *Learning Analytics* (Figura 3). El ciclo presentado por Doug Clow se divide en 4 pasos:

- **Estudiantes:** Es el punto inicial del proceso, analizar a los estudiantes involucrados en el proceso educativo, así como las particularidades del entorno educativo. Dependiendo del tipo de estudiantes y del entorno educativo (estudiantes en un curso universitario, un curso MOOC, participantes de una conferencia de investigación, etc) cambiarán las formas de aplicar el resto de pasos del ciclo.
- **Datos:** La segunda etapa será la generación y captura de datos sobre los estudiantes o el proceso educativo. Algunos de estos datos podrán ser recogidos automáticamente, y otros necesitarán de un equipo dedicado a ello. La clase de datos que se capturen dependerá en gran parte de las conclusiones obtenidas en el paso anterior, el análisis de los estudiantes y del entorno educativo.
- **Métricas:** A partir de los datos obtenidos en el paso anterior, se realizarán métricas y analíticas que proporcionaran una visión general del proceso educativo. Estas métricas incluirán visualizaciones, *dashboards*, listas de alumnos en riesgo, comparaciones entre alumnos, recomendaciones, etc. Al igual que en el paso anterior, muchas de estas analíticas podrán realizarse de forma automática, pero también es posible que se necesite un equipo dedicado a realizar los análisis y sacar las conclusiones a partir de los datos. Este paso, es el más importante de la mayoría de proyectos de análisis, y ha sido el objetivo de muchos estudios y herramientas innovadoras en los últimos años.
- **Intervenciones:** El paso que completa el ciclo es el uso de las métricas para realizar una o más intervenciones tanto en los alumnos como en el proceso educativo. Este paso puede variar mucho en la forma, desde un *dashboard* para que los estudiantes comparen sus progresos con el resto, hasta la intervención directa de los instructores contactando personalmente con algún estudiante que se ha categorizado como en riesgo.

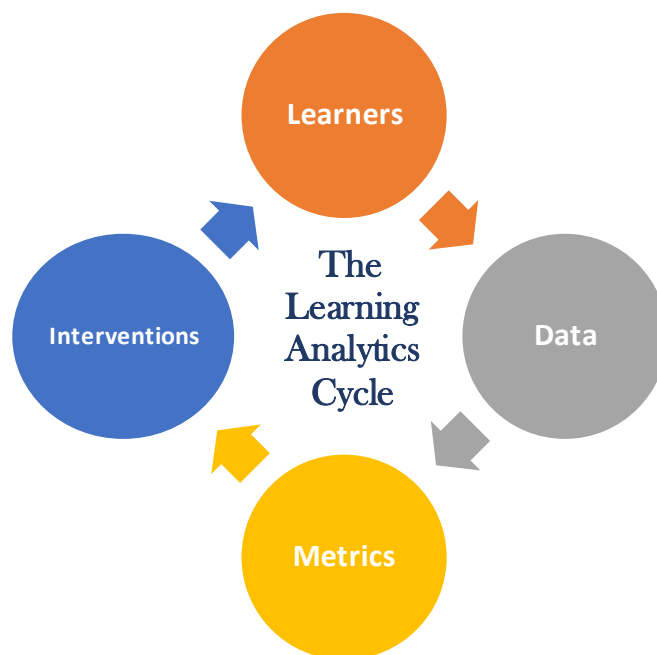


Figura 3: Ciclo de Learning Analytics, por Clow, 2012

El autor hace especial hincapié en el último paso del ciclo, las intervenciones, ya que considera que es el objetivo final del proceso analítico, y que en muchas ocasiones se deja de lado. También comenta que el cumplimiento de todos los pasos no es obligatorio para un proceso de *Learning Analytics*, pero que es posible que la ausencia de uno de ellos haga el proceso mucho menos efectivo.

Learning Analytics en Plataformas MOOC

Una de las características de los cursos MOOC es su carácter masivo, es decir, los cursos están planteados de forma que puedan participar en ellos un gran número de estudiantes. Esto significa que la cantidad de datos generados por los usuarios y que pueden ser obtenidos por las plataformas, será muchísimo más grande que en un curso tradicional. Esta gran cantidad de datos, por una parte permite obtener estadísticas mucho más complejas y fiables que en curso tradicional, ya que la cantidad de muestras será mucho más amplia, pero por otra parte estos datos serán muy difíciles de interpretar sin un correcto sistema estadístico, resultando en muchos casos prácticamente imposible analizar los datos en bruto por los propios encargados de la plataforma o instructores del curso.

Por tanto, la aplicación de técnicas de *Learning Analytics* en plataformas MOOC puede ofrecer resultados muy positivos, tanto para mejorar la calidad de estos cursos, como para el estudio de los procesos educativos en general:

- En los cursos tradicionales, con pocos alumnos, en muchos casos no era necesario el uso de técnicas estadísticas y métricas para que los profesores tuvieran una idea general del funcionamiento del curso. En el caso de los MOOC, sin herramientas estadísticas de alto nivel, puede ser muy complicado y lento sacar conclusiones a partir de los datos en bruto obtenidos de las interacciones de los alumnos. Por tanto la aplicación de técnicas de *learning analytics*, pueden ofrecer análisis que ayuden a los instructores a localizar y mejorar los puntos débiles del curso y fomentar los puntos fuertes.

- La tasa de abandono en los cursos MOOC es mucho más alta que en los cursos tradicionales[23]. Entre otras razones, detectar posibles alumnos en riesgo es mucho más complicado que en curso tradicional con pocos alumnos, por tanto la aplicación de técnicas de *learning analytics* para la detección de este tipo de alumnos, puede ser de una gran utilidad, tanto para detectar y atacar las posibles razones de estos abandonos, como para abordar individualmente cada uno de los casos.
- Por último, el carácter masivo de los cursos MOOC, puede ser muy útil a la hora de obtener estadísticas mucho más fiables que de un curso tradicional, debido a la gran cantidad de muestras. Estas estadísticas además de servir para obtener una visión tanto general como individual del curso y sus alumnos, pueden ser de gran utilidad para realizar todo tipo de estudios orientados a mejorar los procesos educativos, no sólo en MOOC o e-learning, sino prácticamente para todos los ámbitos a los que llega la educación.

A pesar de las evidentes ventajas que la aplicación de estas técnicas traería tanto a las plataformas MOOC como a la comunidad educativa, la mayoría de las plataformas MOOC aún no han implementado estas técnicas, o solo las han implementado de forma parcial. Esto es debido, en gran parte, a que estas plataformas aún son muy nuevas, y la implementación de estas técnicas no es trabajo fácil. Por otra parte, también son muy escasos los estudios sobre la aplicación de *Learning Analytics* en plataformas MOOC, en parte por la reciente aparición de estas, pero también por el carácter propietario de gran parte de las plataformas.

Learning Analytics en EDX

La plataforma Open edX, a pesar de ser una de las plataformas MOOC con más éxito en la actualidad, es también una de las más recientes. Debido a esto, las técnicas de *learning analytics* implementadas, aunque existentes, son muy limitadas.

Por una parte, dentro de los módulos integrados en la plataforma edX se encuentran dos apartados que hacen uso en cierta medida de técnicas de *learning analytics*, uno para la consulta de los alumnos, y otra para el uso de los instructores del curso y el staff de la plataforma.

- **Dashboard alumno**

La pestaña *Progress* de la plataforma mostrara al alumno un resumen de su actividad y progreso en el curso en el que se encuentre.

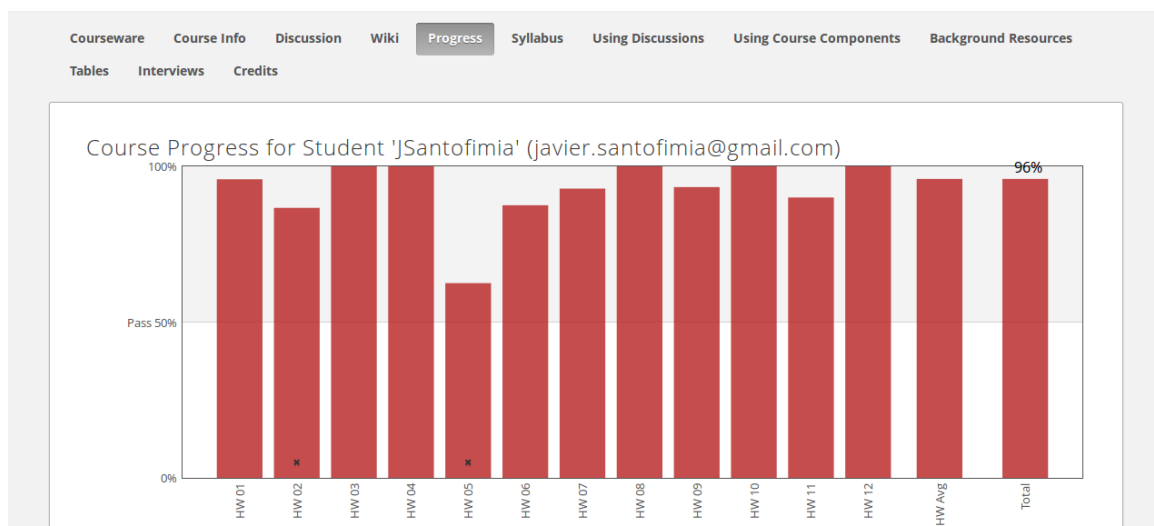


Figura 4: Calificaciones de todas las secciones evaluadas

Por una parte se presenta una gráfica como la de la Figura 4, con las calificaciones de todas las secciones evaluadas, tanto para ejercicios, laboratorios como para los exámenes realizados, incluyendo notas finales, secciones descartadas, etc. Por otra parte se ofrece el desglose de las calificaciones de cada uno de los capítulos del curso como se puede ver en la Figura 5.

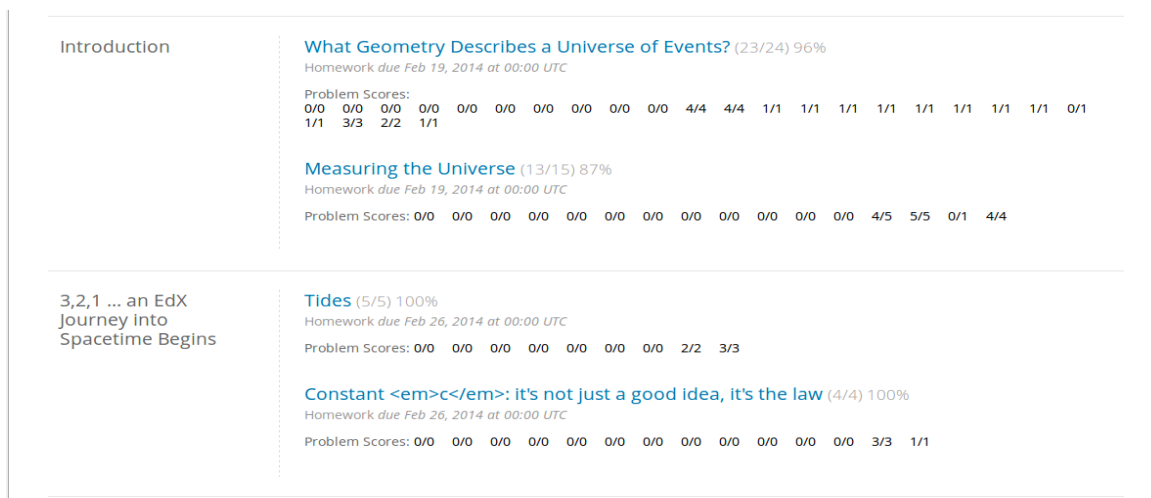


Figura 5 Desglose de las calificaciones de cada capítulo del curso

Como se puede ver, los datos que la plataforma ofrece al alumno se reducen a las calificaciones obtenidas a lo largo del curso.

- **Dashboard instructor**

Por su parte, los instructores de los cursos, cuentan con una pestaña *Instructor* para gestionar múltiples aspectos del curso como la gestión de usuarios del curso (añadir, eliminar, permisos, etc), ajustes de notas de estudiantes, o gestión de e-mails.

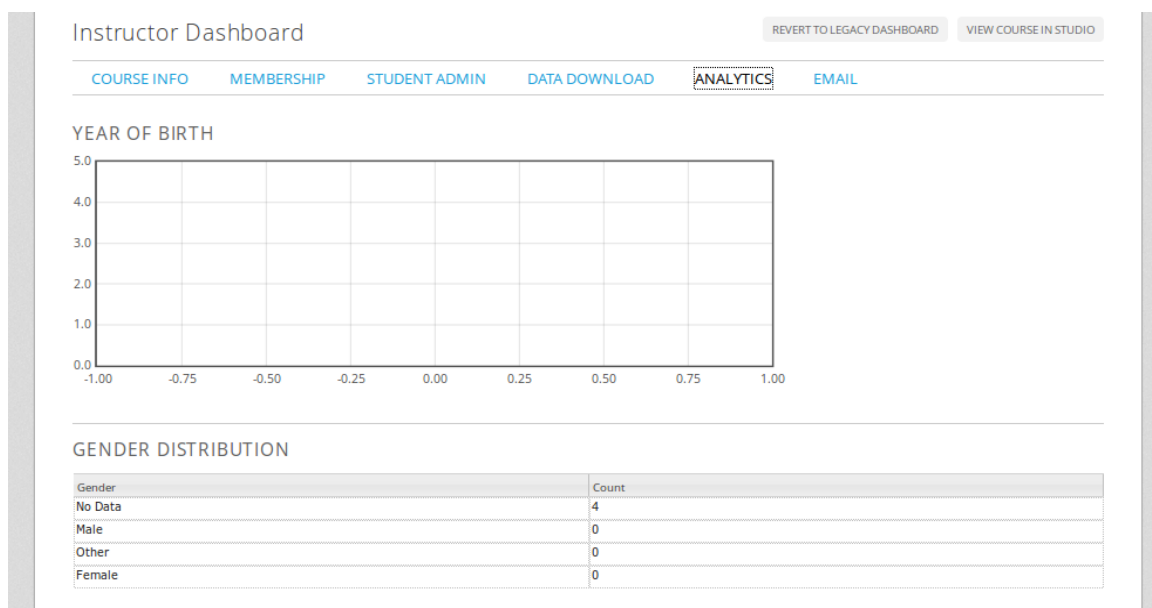


Figura 6: Analytics - Estadísticas del Curso

Entre estos ajustes también se pueden encontrar una pestaña llamada *Analytics* (Figura 6) que ofrece algunas estadísticas sobre el curso, aunque por ahora las estadísticas se reducen a mostrar la distribución de edad, sexo, y nivel de educación de los alumnos.

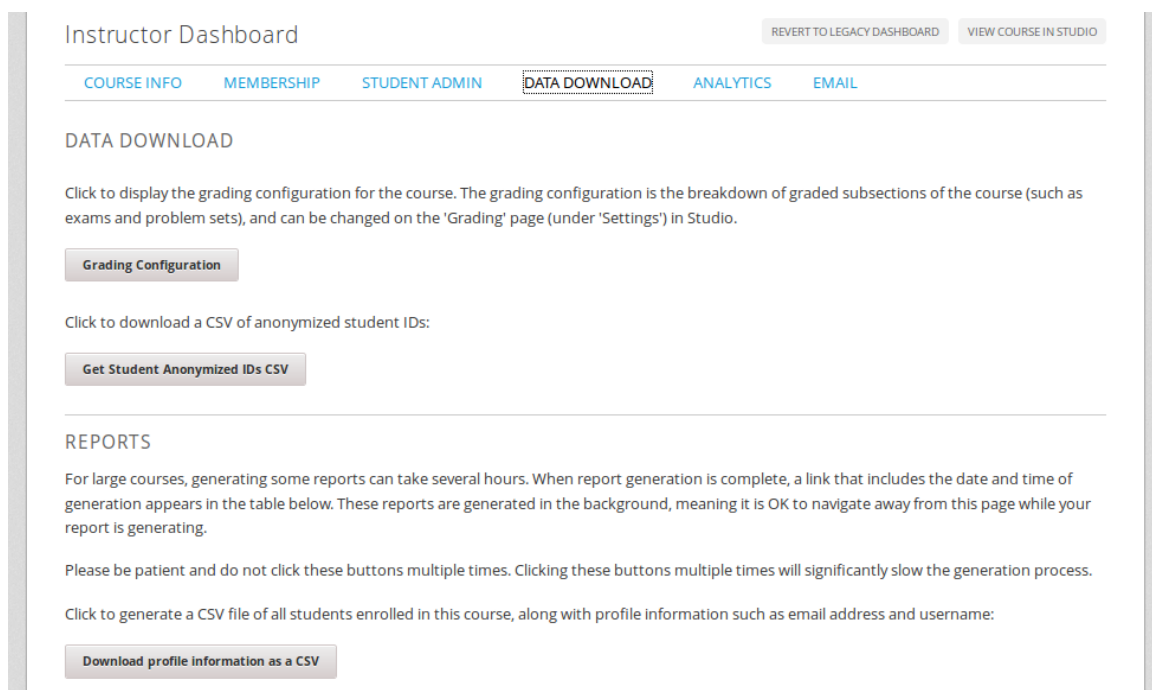


Figura 7: Data Download del Dashboard de instructor

Por otra parte, en la pestaña *Data Download* (Figura 7) los instructores pueden generar algunos informes en formato CSV para tratar los datos externamente. Aunque una vez más, la información que se puede obtener es escasa: tan solo se puede obtener una lista de los alumnos con sus datos personales (email, origen, sexo, etc).

Como se puede ver, el soporte de *Learning Analytics* que ofrece en la actualidad la plataforma Open edX de forma nativa, aunque útil, es escaso comparado con las posibilidades que podrían ofrecerse. Especialmente insuficientes, son las estadísticas con las que cuentan los instructores, ya que con lo ofrecido en su *dashboard*, es muy complicado que puedan aplicar la última fase del ciclo de *learning analytics*: las intervenciones.

Además de las estadísticas integradas oficialmente en la plataforma, también se encuentran algunos proyectos externos a la plataforma, que buscan ofrecer datos a usuarios o a terceras partes, para realizar análisis de estos con múltiples objetivos.

- **Open edX Analytics**

Es un proyecto de la comunidad edX encargado de la gestión de datos de la plataforma. El objetivo es ofrecer grandes cantidades de datos de los cursos a instructores (los que pueden obtener en la *dashboard* del instructor, como se ha comentado anteriormente), y principalmente a las instituciones asociadas que ofrezcan cursos en la plataforma, para el posterior análisis de ellos. El objetivo, es que estos análisis sirvan para mejorar la experiencia educativa, tanto en los cursos que se analizan como en el resto de cursos.

Debido a los problemas de confidencialidad que puede acarrear la entrega de estos datos, las instituciones que quieran tener acceso a estos datos tendrán que nombrar una o varias personas, llamadas *Data Czar*, que serán los responsables de acceder a estos datos y entregarlos de forma segura a las instituciones para su análisis. Estos *Data Czar*, podrán acceder a los datos de los cursos de sus respectivas instituciones, que se encuentran almacenados de forma segura y encriptada en la nube, empleando Amazon Simple Storage Service (Amazon S3).

Los datos que se ofrecen incluyen datos de los alumnos y su progreso, datos generales del curso, así como datos de las discusiones de los foros y de la *wiki* del curso. Estos datos se entregan en dos paquetes distintos, por una parte se ofrecen los volcados de datos de las tablas de las bases de datos del curso, y por otra, un *log* con los eventos producidos por los usuarios del curso registrados por la plataforma edX. Los datos que se ofrecen se actualizan semanalmente en el caso de los volcados de bases de datos, y diariamente en el caso de los eventos.

- **edX Insights:**

EdX Insights (*Figura 8*) es un módulo para la plataforma edX, desarrollado por la comunidad y lanzado en Octubre de 2014 [24]. El objetivo del proyecto es por una parte, proporcionar información de los cursos a los instructores de estos, y por otra ofrecer un API a la comunidad para poder desarrollar nuevas estadísticas.

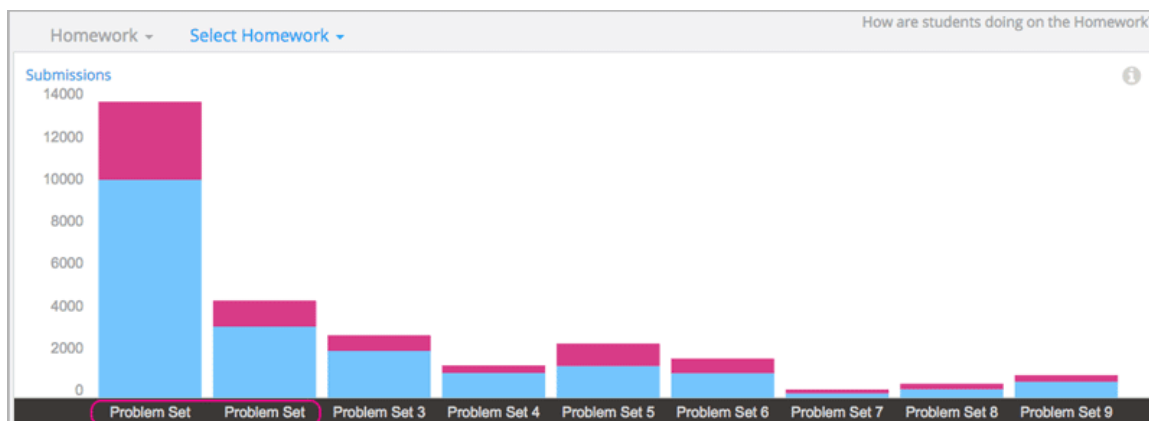


Figura 8: Captura de una de las visualizaciones de edX Insights

El módulo no está integrado con la plataforma edX, y actualmente se encuentra alojado de forma externa y funcionando para cursos alojados en la rama edge de edX, es decir, la rama que aloja los cursos no oficiales de edX.

Las estadísticas que el módulo presenta en la actualidad se dividen en tres ramas diferenciadas:

- **Inscripciones:** En este menú se presentan diversas estadísticas y visualizaciones relacionadas con la inscripción en el curso, así como estadísticas demográficas sobre estas inscripciones.
- **Actividad de los estudiantes:** En esta segunda rama se ofrecen estadísticas relacionadas con la actividad de los estudiantes en el curso, como el número de estudiantes que han visto un vídeo o han realizado un ejercicio.
- **Rendimiento de los estudiantes:** En este último menú se pueden consultar los resultados que tienen los estudiantes al contestar cada uno de los posibles ejercicios presentes en el curso, tanto para ejercicios evaluables como no evaluables.

El módulo, debido a su reciente aparición y estado de desarrollo, aún no presenta estadísticas de excesiva complejidad, pero muchas de ellas complementan a las que podemos encontrar integradas en la plataforma y pueden ser de mucha utilidad.

En general, se puede observar que el uso de técnicas de *learning analytics* es muy escaso en la plataforma, y aunque el grueso de datos que ofrece la plataforma para análisis externos es muy grande, no existe una herramienta integrada que ofrezca las ventajas que el uso de estas técnicas puede dar tanto a alumnos como a instructores del curso.

2.3. TECNOLOGÍAS EMPLEADAS

En este apartado se hace un repaso a las tecnologías que se emplean para el desarrollo de la plataforma edX y a las nuevas tecnologías que se incluirán para el desarrollo del módulo de *Learning Analytics*.

2.3.1. Tecnologías empleadas en el desarrollo de Open edX

El proyecto Open edX tiene un tamaño considerable, por lo que se usan multitud de tecnologías, lenguajes de programación y bases de datos.

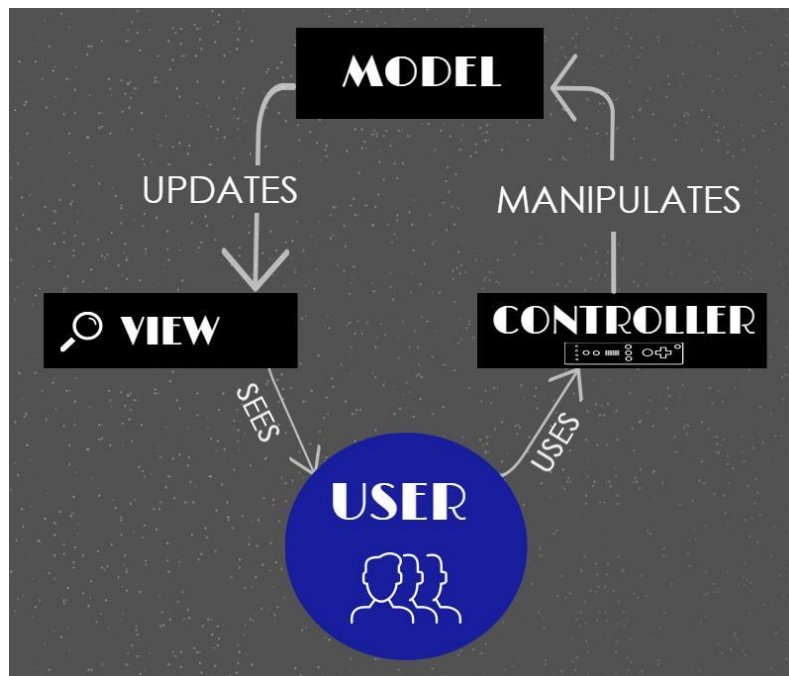


Figura 9: Patrón Modelo-Vista-Controlador

La principal tecnología empleada en el desarrollo de Open edX es el *framework* web Django. Django es un framework de desarrollo web de código abierto mantenido por la *Django Software Foundation*. El objetivo principal de Django es facilitar el desarrollo de sitios web complejos, facilitando la reusabilidad y extensión de componentes, el desarrollo rápido y el principio DRY (*Don't Repeat Yourself*). Django está escrito en Python y su arquitectura se basa en el patrón Modelo-Vista-Controlador. Este patrón de arquitectura software, como se puede observar en la Figura 9, se basa en 3 componentes: el modelo, que representa la información y gestiona el acceso a esta, el *controlador*, que realiza peticiones al *modelo* a raíz de eventos (generados en su mayor parte por el usuario), y la *vista*, que presenta la información del *modelo* al usuario.

El desarrollo de Open edX está basado en Django, siendo por ejemplo, el LMS y el CMS sendos proyectos Django. A continuación se presentan las tecnologías más importantes que se emplean, siguiendo el patrón MVC:

- **Modelo:** Para la implementación del modelo se emplea principalmente una base de datos relacional SQL, MySQL o SQLite según el tipo de instalación de la plataforma. Esto está motivado por la buena integración de Django con estas bases de datos, permitiendo que la manipulación de estas desde el código de la parte controlador sea muy sencilla y cómoda. Para ciertos datos (en general datos de gran tamaño) también se emplea una base de datos MongoDB, a la que es posible acceder directamente o mediante librerías intermedias. MongoDB es un sistema de código abierto de bases NoSQL, es decir, en lugar de guardar los datos en tablas, como hacen las bases

relacionales, guarda estructuras de datos en un formato llamado BSON, basado en el formato JSON.

- **Vista:** Para la parte vista del patrón, Open edX emplea el sistema de plantillas Mako, que permite introducir código Python para generar dinámicamente el código HTML que se mostrará. Para completar Mako y HTML, también se emplea CSS para el diseño y ficheros JavaScript para múltiples funciones.
- **Controlador:** En su mayor parte, la parte controlador está codificada en Python, aunque hay partes y librerías escritas en otros lenguajes, pero no serán analizadas en profundidad ya que no son relevantes para el desarrollo de este proyecto. Muchos de los ficheros JavaScript empleados en la parte de vista también realizan funciones atribuibles a la parte controlador del patrón.

2.3.2. Nuevas tecnologías introducidas para el desarrollo de un módulo de learning analytics

En general, para el desarrollo del módulo de *learning analytics* para Open edX, intentaremos emplear las mismas tecnologías y metodologías que se emplean para el desarrollo de Open edX. Pero habrá ciertos aspectos de nuestro desarrollo que se simplificarán mucho haciendo uso de tecnologías o librerías externas, no empleadas anteriormente por los desarrolladores de edX.

Scheduler: Celery beat

Celery [25] es un encolador de tareas basado en el paso de mensajes distribuidos. Celery está enfocado en el procesamiento de tareas en tiempo real, y es empleado en el desarrollo de Open edX con este objetivo para varias tareas, como el envío masivo de emails o para el proceso de calificaciones de tareas. Pero también es posible emplear Celery como *scheduler*, con su componente *celery beat*.

Debido a la gran carga de datos que pueden tener los cursos MOOC, necesitaremos planificar el cálculo de muchas de las estadísticas que se presentarán en nuestro módulo. Para ello haremos uso del componente *celery beat* del encolador de tareas Celery, ya que la integración en la plataforma Open edX será sencilla, al ya emplear estos otros componentes de Celery.

Visualizaciones: Google Charts

Uno de los objetivos del proyecto es que, una vez obtenidos los datos y estadísticas, estos se presenten a los usuarios de forma sencilla, para esto, muchos de ellos se presentarán de forma gráfica. En el desarrollo de Open edX, las gráficas que se muestran no emplean ninguna librería externa, y son desarrollados en JavaScript (como por ejemplo el gráfico de progreso comentado en la sección X). Esto puede ser una buena opción cuando se necesita dibujar pocas gráficas, pero en nuestro caso es preferible usar una librería externa que nos proporcione una forma sencilla y unificada de incluir gráficas en nuestra *dashboard*. Para esta tarea, se ha decidido emplear la librería de Google, Google Charts. [26]

Google Charts es una herramienta mantenida por Google, que nos permite incluir gráficas complejas en JavaScript. Ofrece multitud de tipos de gráficas, con un soporte para todo tipo de navegadores web y una sencilla personalización de estas, así como la posibilidad de incluir interactividad en ellas.



CAPÍTULO 3

3. ARQUITECTURA

En este capítulo se presenta el diseño de la arquitectura del módulo de *learning analytics* que se va a implementar. Se presentan los componentes de cada uno de los bloques del proyecto, el modelo (bases de datos), la vista, y el controlador (API de datos, cálculo de estadísticas y tareas periódicas). Antes de poder presentar la arquitectura del módulo, es necesario realizar un análisis de la arquitectura de la plataforma Open edX, y especialmente, de los componentes de esta con los que interactuarán el módulo que se está implementando en este proyecto.

3.1. ARQUITECTURA OPEN EDX

En esta sección se hace un análisis general a la arquitectura del proyecto Open edX, centrándose en los módulos o aplicaciones más relacionadas con este proyecto, y haciendo un análisis más profundo en el almacenamiento y gestión de datos registrados por la plataforma. En la *Figura 10* se puede ver una arquitectura general de la plataforma.

Como se comentó en el capítulo anterior, la plataforma Open edX está desarrollada a partir del *framework* web Django, y emplea la estructura definida por Django para la mayoría de aplicaciones y módulos. Gran parte de la gestión de bases de datos también se hace a través de Django.

Aunque también hay muchos módulos y librerías externas al *framework* web Django. Se estudiarán las que tengan mayor relevancia, así como las que sean necesarias para el funcionamiento del módulo de *learning analytics*.

El grueso de edX está compuesto por dos aplicaciones principales: Content Management System (CMS) y Learning Management System (LMS). El primero es una herramienta de creación de cursos, en la que los instructores diseñan e incluyen los contenidos de los cursos que van a ofrecer. El LMS es el espacio que usan tanto los alumnos como los instructores para realizar los cursos.

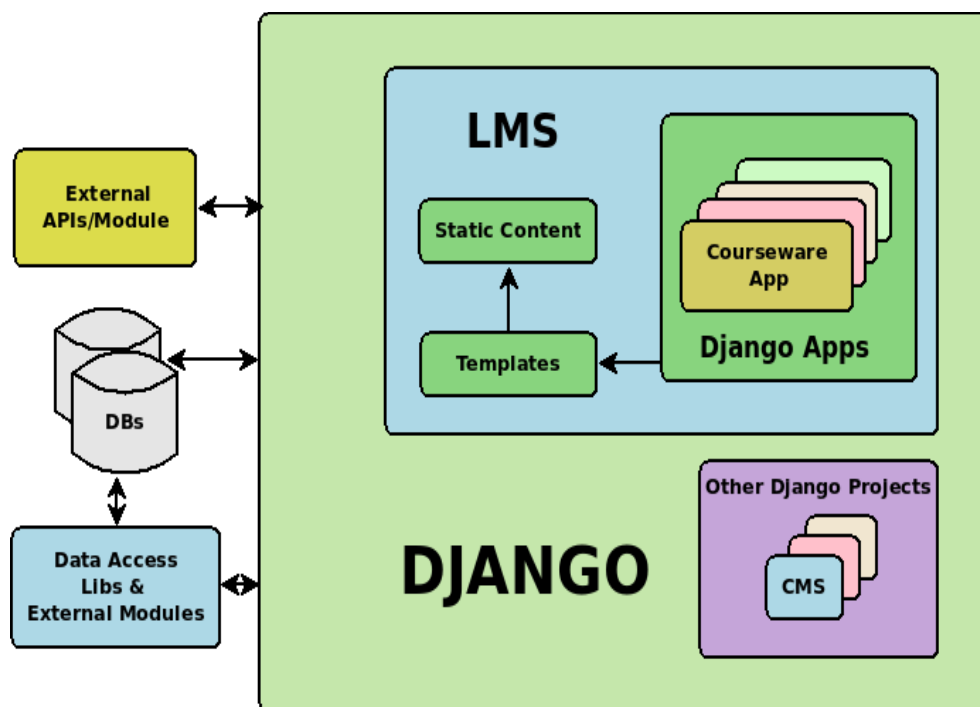


Figura 10: Arquitectura general de Open edX

Dado que este proyecto va incluido en el LMS y no emplea funcionalidades del CMS, el estudio de la plataforma se centra en el LMS y en las librerías y módulos que este emplean.

3.1.1. Arquitectura del LMS

La estructura del LMS es la estructura típica de un proyecto Django: una serie de aplicaciones Django, escritas en su mayor parte en Python, que contienen la funcionalidad y los modelos (Controlador y Modelo del paradigma MVC). Las aplicaciones, además de emplear las utilidades que ofrece el *framework* web Django, harán uso de otras librerías y módulos externos, de los cuales se analizan los más relevantes para este proyecto en otros apartados de este capítulo. Estas aplicaciones hacen uso de plantillas, o *templates*, para presentar los datos de los modelos. Como se ha comentado en el capítulo anterior, a pesar de que Django proporciona su propio lenguaje de *templates*, en el proyecto Open edX se usa el lenguaje de *templates* Mako en la mayor parte de estas. Estas plantillas a su vez usaran archivos estáticos (principalmente JavaScript y CSS) incluidos también dentro del proyecto del LMS.

El módulo de *learning analytics* es una aplicación Django más dentro del LMS, con sus respectivas *templates* y archivos estáticos, y que al igual que el resto de aplicaciones, hará uso de las funcionalidades de Django y de las librerías y módulos externos a Django del proyecto Open edX.

3.1.2. Courseware (Cursos)

Los cursos de la plataforma Open edX son un conjunto de módulos individuales que se combinan jerárquicamente. La estructura típica de un curso en la plataforma se muestra en la (Figura 11). Cada uno de estos módulos será un capítulo, una sección, un problema, un vídeo, etc. Estos módulos se modelan a través de unas *APIs* diseñadas por los desarrolladores de Open edX: xModules y xBlocks.

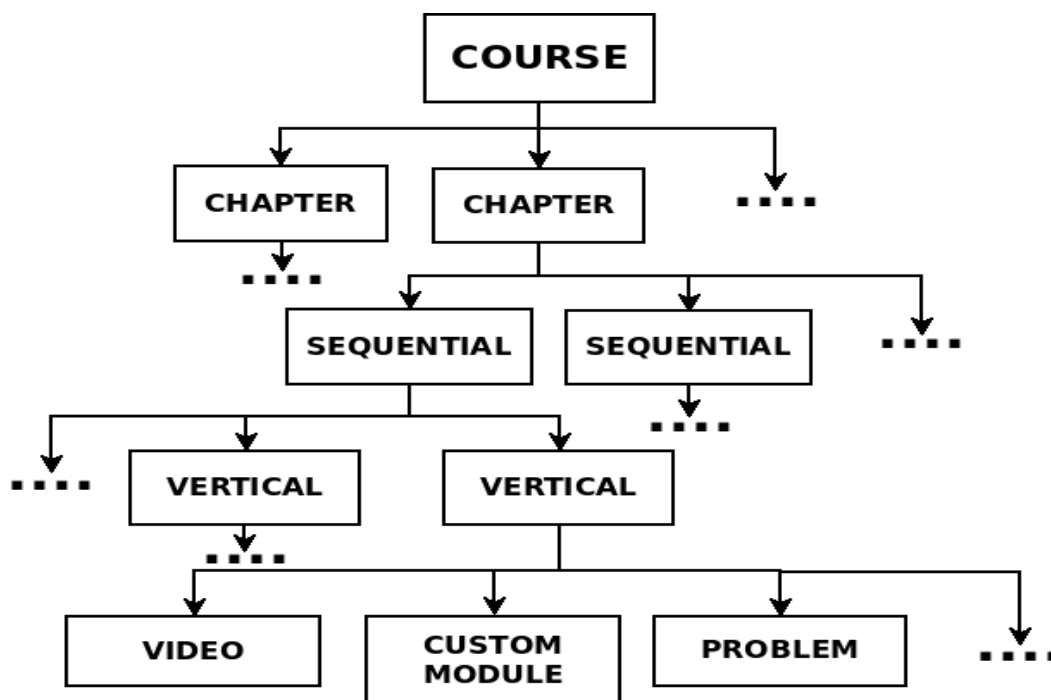


Figura 11: Estructura de los cursos en Open edX

xModule es común al LMS y el CMS de Open edX y ofrece cada uno de los módulos del *courseware* (capítulos, problemas, vídeos, etc) como un objeto Python. Estos xModules no funcionan por sí mismos, sino que corren dentro de aplicaciones web, como pueden ser el LMS y el CMS, que prestan servicios a los módulos, como puede ser el acceso y modificación de las bases de datos del sistema, mapeo de URL o autenticación. Además de ofrecer los datos de estos módulos, los objetos xModules, también tienen funciones para realizar gestiones sobre estos objetos indicando un contexto (es decir, un usuario haciendo uso de un curso), tales como poner nota a un problema o recalcular la nota media de un curso.

XBlock es la evolución de xModule. Surge para mejorar la funcionalidad de xModule. El principal objetivo es facilitar la creación de nuevos módulos (XBlocks) por parte de desarrolladores externos, pero también tiene como objetivo mejorar el diseño y la arquitectura de la anterior API.

En la actualidad la API XBlock se encuentra en desarrollo, y el uso de esta en la plataforma no es aún estable ni completo. Aunque ciertas partes de la plataforma ya se encuentran migradas a XBlock, para el desarrollo de este proyecto, se hará uso de la API antigua, xModule.

3.1.3. Gestión de datos: Bases de datos

Open edX emplea al menos tres formas distintas para almacenar y consultar datos de forma persistente (Figura 12). Por una parte, la plataforma cuenta una base de datos relacional que varía de tecnología según la implementación de la plataforma que se emplee (para desarrollo o para producción). La plataforma Open edX, también cuenta con una base no relacional Mongo DB. Por último, algunos datos también se almacenan en archivos, normalmente en formato XML. Para acceder a estos datos se emplearán los modelos de Django en el caso de la base de datos relacional, y los módulos de la plataforma (ya sean xModules o xBlocks) para obtener los datos de Mongo DB y los ficheros.

- **Base de datos relacional (Modelos Django)**

La base de datos relacional, es el principal elemento del modelo (MVC) de un proyecto Django, debido a las facilidades que el *framework* ofrece al desarrollador para manejar los datos en estas bases. Django puede funcionar con varios tipos de bases de datos relacionales (PostgreSQL, MySQL, SQLite, Oracle) [27] y ofrece una capa de abstracción con respecto a estas a través de su gestión de *Models*.

En el caso de la rama actual de Open edX, solo se soporta oficialmente el uso de base de datos MySQL o SQLite para la instalación para desarrollo, y MySQL para la versión en producción. En estas bases de datos, se almacenan datos de usuarios sencillos y no muy pesados. El acceso y gestión de estos datos se realiza de forma sencilla gracias a las herramientas ofrecidas por Django.

- **Base de datos no relacional y ficheros**

Para datos más pesados, como el contenido de los módulos de los cursos o de los foros de discusión (es decir, *contentstore*), se emplea la base de datos no relacional MongoDB. Al principio del desarrollo de Open edX, estos datos se almacenaban en ficheros, comúnmente XML, pero en la actualidad la mayoría de datos están migrados a Mongo DB.

Aún quedan algunos módulos o partes del código que guardan los datos en ficheros, pero esto no es un problema para el desarrollo de este proyecto, ya que se intenta acceder a estos datos empleando las *APIs* xModule o XBlocks, que funcionan como capas de abstracción con respecto al origen de los datos. Gracias al uso de estas APIs es posible manejar los datos de Mongo DB y de los ficheros XML, sin tener que acceder directamente a ellos, lo cual es más sencillo a la hora de desarrollar, y sobre todo, a la hora de mantener el código, ya que un cambio en la forma de almacenar los datos, o en algún campo de estos, posiblemente no repercuta en el código al actualizar el API de xModule y XBlock.

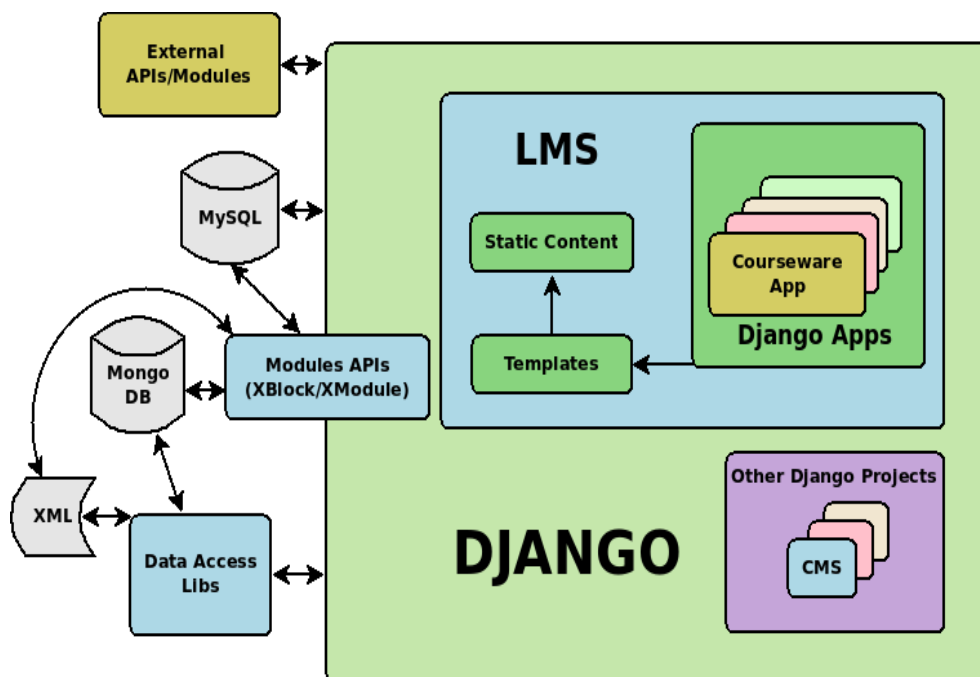


Figura 12: Bases de datos en la arquitectura de Open edX

Por último, algunas partes de la plataforma Open edX acceden directamente a las bases de datos, ya sea por llamadas SQL en el caso de la base de datos relacional, o mediante librerías de Python, como PyMongo, para el caso de Mongo DB. Aunque es una forma correcta de acceder a las bases de datos, en este proyecto se intenta no usar accesos directos a ellas, y emplear las herramientas existentes (Modelos de Django y xModule/xBlocks) por sencillez de desarrollo y de mantenimiento del código en el futuro.

3.1.4. Gestión de datos: Tracking Logs

La plataforma Open edX recoge muchas de las acciones que realizan los usuarios sobre esta, almacenándolas en un *log* en forma de eventos [28]. Estos eventos se almacenan en un fichero de texto para consulta de los administradores de la plataforma, pero también es posible indicar a la plataforma que los almacene en la base de datos SQL, haciendo *true* la variable global `'ENABLE_SQL_TRACKING_LOGS'` en los ficheros de configuración de entorno del LMS.

Los eventos recogidos se dividen en eventos de los estudiantes, es decir cualquier evento que ocurra fuera de la *dashboard* de instructor, y los de instructor, eventos que ocurren dentro de dicha *dashboard*. Todos los eventos tienen unos campos comunes que indican información general del evento ocurrido como puede ser el origen del evento (servidor o cliente), el tiempo en el que ha sucedido o un contexto general del evento (curso, usuario, etc).

Entre los eventos de estudiantes que se almacenan se encuentran algunos de mucha importancia para la obtención de datos para el cálculo de estadísticas para el módulo de *learning analytics*:

- **Inscripción de usuarios:** se generan eventos cada vez que un usuario se inscriba o se borre de un curso, así como cuando cambie la configuración de una inscripción previa.
- **Eventos de navegación:** Estos eventos se generan cuando un usuario navega dentro del curso, e indican a que parte del curso ha navegado o si ha cerrado alguna pagina de este.
- **Interacción con vídeos:** Indican las interacciones de los usuarios con los vídeos del curso: cuando empiezan a verlo, pausas, cambios de velocidad, navegación dentro del vídeo, activación/desactivación de subtítulos, etc.
- **Interacción con problemas:** También se registran gran parte de las interacciones de los usuarios y el servidor con los problemas del curso: cuando se muestra un problema, cuando el usuario contesta a estos, el cálculo de la nota por parte del servidor, petición de la respuesta por parte del usuario, etc.
- **Otros eventos:** Además de los eventos expuestos, se registran otros muchos, aunque no de tanta utilidad para las estadísticas del módulo de *learning analytics* que se desarrollan en este proyecto: interacción de los usuarios con libros de texto o librerías de contenido, eventos generados en los foros de discusión, evaluaciones de ejercicios por parte de los estudiantes o eventos generados por aplicaciones de terceras partes.

Por otra parte los eventos de instructor registrarán algunas de las posibles interacciones de los instructores con la *dashboard* de instructor: registro/expulsión de instructores, gestión de los administradores de los foros de discusión, re-evaluación de los problemas del curso, peticiones de información del curso por parte de los instructores, etc.

Los eventos recogidos en los *tracking logs* son de gran utilidad para el modulo que se diseña en este proyecto, por lo que el módulo activará la opción de configuración comentada anteriormente para que estos eventos sean guardados en la base de datos. Con los eventos almacenados en la base de datos SQL será posible realizar peticiones complejas de información de forma rápida y sencilla, aunque repercutirá en el tamaño de esta base de datos de forma significativa, debido a la gran cantidad de eventos que se registran.

3.2. ARQUITECTURA INTERNA DEL MÓDULO DE LEARNING ANALYTICS

En esta sección se presenta la arquitectura del módulo de *learning analytics* que se desarrolla en este proyecto. Primero se muestra el lugar del módulo dentro de la arquitectura del proyecto Open edX. Posteriormente se analiza la arquitectura del módulo y se hace un repaso de las partes más importantes de este.

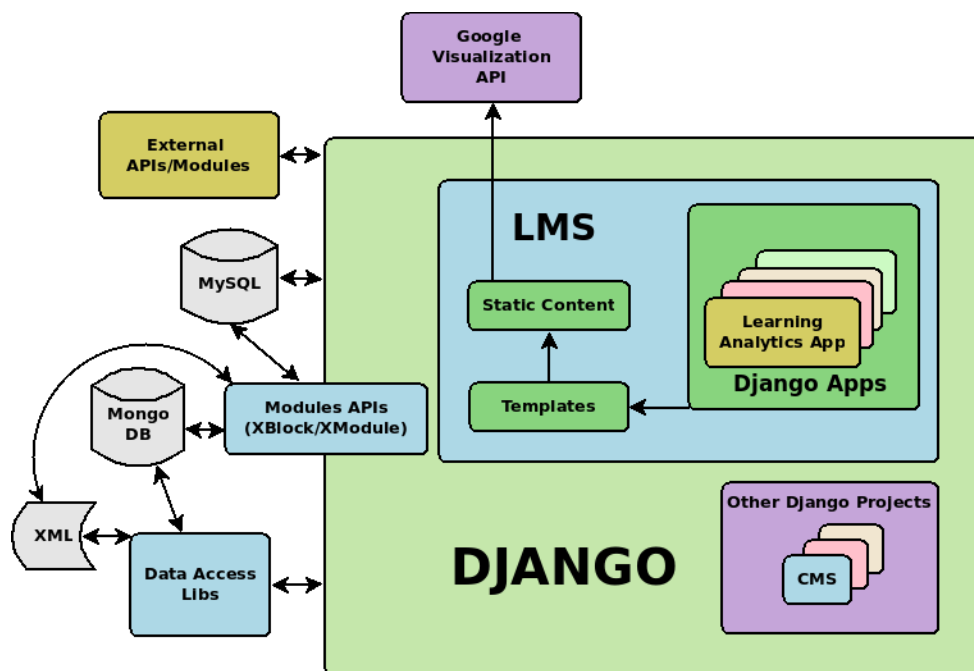


Figura 13: Módulo de learning analytics en la arquitectura de Open edX

Como se puede ver en la Figura 13, el módulo de *learning analytics* se encuentra dentro del proyecto LMS de la plataforma Open edX, como una aplicación Django más.

3.2.1. Arquitectura

La aplicación de *learning analytics* para el LMS, sigue la arquitectura típica de una aplicación Django, siguiendo el patrón MVC. Para el diseño e integración de este módulo en la plataforma Open edX se ha intentado ser consecuente con las tecnologías, librerías y metodologías empleadas en el resto de la plataforma por los desarrolladores de esta.

Por un lado en el fichero *models.py* se definen los modelos, o tablas de la aplicación en la base de datos, que serán donde se almacenen las estadísticas a calcular, para que puedan ser consultadas rápidamente cuando un usuario haga una petición.

La vista de la aplicación se gestionará, al igual que en la mayoría de aplicaciones Django, mediante el fichero *views.py*, que, cuando reciba una petición por parte de un usuario, enviará los cálculos de la base de datos a las plantillas y se encargará de obtener nuevos datos y volver a enviarlos mediante AJAX cuando los usuarios interactúen con las visualizaciones. Estas plantillas a su vez harán uso de ficheros estáticos, principalmente ficheros JavaScript y CSS para mostrar las visualizaciones.

Por último, la parte controlador de la aplicación, se encargará de calcular las estadísticas y almacenarlas en la base de datos. Esta parte compone el grueso del código de la aplicación, y esta separada en tres partes diferenciadas:

- **API de datos**

Como se comenta en la introducción de esta memoria, uno de los objetivos del proyecto es crear un API para obtener datos relevantes de la plataforma Open edX. Este API tiene como

objetivo facilitar la obtención de datos para el posterior cálculo de estadísticas para el módulo de *learning analytics*, pero también está pensado para que pueda ser empleado o ampliado por cualquier desarrollador que necesite hacer uso de él. El código de este API se encuentra en el fichero *data.py*, y sus funciones se analizan en profundidad en el siguiente capítulo.

- **Cálculo de estadísticas**

Haciendo uso del API diseñada, y de las herramientas que ofrece Django y las librerías de Open edX, se calculan las estadísticas necesarias para cada una de las visualizaciones que se presentarán en el módulo de *learning analytics*. Esta parte, incluirá funciones que se encargan de calcular cada una de las estadísticas y de actualizar el contenido de la base de datos con los resultados. También se incluyen funciones auxiliares para el cálculo de las estadísticas, que debido a que su aplicación es muy específica al cálculo de las estadísticas del módulo, se ha decidido no incluir en el API de obtención de datos. El código de este bloque se encuentra en el fichero *analytics.py* y se analiza en detalle en el siguiente Capítulo 5.

- **Tareas periódicas**

Por último, es necesario gestionar el cálculo asíncrono de las estadísticas, de forma que no saturan el servidor o el cliente al realizar las peticiones. Para este trabajo se empleará Celerybeat, un módulo del gestor de colas Celery, que se encarga de llamar a las funciones incluidas en el fichero *task.py* periódicamente de forma que las estadísticas estén actualizadas cuando sean consultadas.

En los siguientes apartados, se hace un análisis de la estructura y contenido de cada una de estas partes, así como de los modelos de la base de datos y la arquitectura que siguen las plantillas y JavaScripts para presentar las visualizaciones.

3.2.2.Modelo: Tablas en base de datos

Para almacenar los datos de las estadísticas calculadas en el módulo de *learning analytics* se crean varias tablas en la base de datos de la plataforma. Se empleará únicamente la base de datos relacional, ya que los datos que se incluirán son de poco tamaño, es necesario poder hacer consultas complejas de forma rápida y eficaz y gracias a las herramientas y facilidades que ofrece Django es muy sencillo crear y mantener las tablas, así como gestionar estas desde el código.

Los nombres de las tablas creadas para este módulo, empezarán todas por el prefijo *learning_analytics*. A continuación se describen dos de las tablas creadas que serán usadas por muchas de las visualizaciones y por parte de varias funciones de la parte controlador del módulo.

- **CourseStruct**

En esta tabla se almacena la estructura de cada curso. Aunque la plataforma Open edX ofrece varias formas de obtener esta estructura, se ha decidido crear esta tabla para que el acceso a estos datos sea sencillo y rápido, pero sobre todo para que sea posible realizar consultas más complejas, como obtener todos los módulos de un tipo (capítulo, sección, etc)

u obtener directamente el hijo de un módulo. Los campos de la tabla de la base de datos se describen en la Tabla 2.

Tabla 2: *CourseStruct* - Campos de la tabla de la base de datos

Columna	Tipo	Null	Key	Descripción
<i>id</i>	int	NO	PRI	ID de la entrada en la tabla
<i>course_id</i>	CourseKeyField	NO	MUL	ID del curso. Este campo junto con <i>module_id</i> tienen que ser únicos
<i>module_id</i>	LocationKeyField	NO		ID del módulo. Este campo junto con <i>course_id</i> tienen que ser únicos
<i>name</i>	Char	NO		Nombre del módulo
<i>section_type</i>	Char	NO	MUL	Tipo del módulo. Los valores posibles son: “chapter”, “sequential” o “vertical”
<i>index</i>	int	NO		Posición dentro de la jerarquía del curso
<i>father</i>	CourseStruct(Foreign-Key)	YES		El padre del módulo en la jerarquía del curso
<i>graded</i>	boolean	NO		Indica si el módulo es evaluable o no
<i>released</i>	boolean	NO		Indica si el módulo está ya publicado

- **StudentGrades:**

Esta otra tabla almacenará las notas de cada uno de los estudiantes para un curso dado.

Tabla 3: *StudentGrades* - Campos de la tabla de la base de datos

Columna	Tipo	Null	Key	Descripción
<i>id</i>	int	NO	PRI	ID de la entrada en la tabla
<i>student_id</i>	int	NO		ID del estudiante. Este campo junto con <i>course_id</i> tienen que ser únicos
<i>course_id</i>	CourseKey	NO	MUL	ID del curso. Este campo, junto con <i>student_id</i> tienen que ser únicos.
<i>grades</i>	Text	NO		Diccionario con las notas del alumno para cada sección. Se entra en detalle de la forma de este campo en el siguiente capítulo
<i>grade_group</i>	Char	NO		Grupo de alumnos en el que está el alumno según su nota media. Los valores posibles son: “PROF”, “OK” y “FAIL”
<i>last_calc</i>	DateTime	NO		Fecha de la última modificación de la entrada

También indicará en que grupo de alumnos se encuentra el alumno, según su nota media: Proficiency, aprobado o suspenso. Aunque las notas del alumno se emplean únicamente para una visualización, el grupo de alumnos en los que se encuentra el estudiante, es empleado

por la mayoría de las visualizaciones, ya que una de las opciones que ofrecen es mostrar la visualización con los datos de únicamente uno de los grupos. Los campos de la tabla `StudentGrades` se describen en la Tabla 3.

Además de estas dos tablas, se crea una tabla en la base de datos para cada visualización del módulo. Cada una de estas tablas se analiza en detalle en el Capítulo 5 de esta memoria.

3.2.3. Vista: *views.py*, plantillas y static files

La capa vista del módulo contiene todos los elementos que hacen posible al usuario visualizar los datos almacenados en el modelo por la capa controlador.

El primer elemento que actúa en el proceso es el fichero *views.py*. Este fichero contiene dos funciones:

- ***index***: Esta función se encarga de obtener todos los datos necesarios del modelo para poder generar el código HTML de la página del módulo a raíz de la plantilla Mako y dichos datos.

El sistema llamará a la función *index* cuando un usuario realice una petición (es decir seleccione la pestaña «Learning Analytics» de su *courseware*). Por una parte, la función obtiene un contexto de la petición realizada (qué usuario realiza la petición, para qué curso, qué rol tiene este usuario en el curso, etc) y por otra parte, obtiene los datos almacenados en la base de datos de cada una de las visualizaciones que haya que mostrar para el usuario indicado en el contexto (en caso de que el usuario tenga el rol de instructor, obtendrá los datos para todos los usuarios). Dado que, en general, estos datos tendrán la forma de un diccionario Python, la función *index* también formatea estos datos, de forma que los ficheros JavaScript y las plantillas puedan interpretarlos. Generalmente el último paso del formateo de datos es convertir los diccionarios Python a un *String* con formato JSON, fácilmente interpretable por JavaScript. Una vez tiene los datos necesarios, en el formato necesario, los combina con la plantilla Mako mediante la función *render_to_response([Template], [Data])* para obtener el código HTML que mostrará el módulo al usuario.

Esta función tiene el decorador Python *@login_required*, que evita que ningún usuario no autenticado pueda llamarla.

- ***chart_update***: Muchas de las visualizaciones, tienen un selector para que se muestre el conjunto de datos requerido (seleccionar un vídeo en concreto del curso, seleccionar los datos de un alumno en concreto por parte del instructor, etc). Cada vez que el usuario cambia la selección de estos selectores, es necesario obtener un nuevo conjunto de datos de la base de datos. Para realizar esta acción, los ficheros JavaScript que manejan el selector, se encargan de realizar una petición AJAX a la función *chart_update*.

Esta función se encarga de recibir e interpretar la petición AJAX, de obtener los datos necesarios y enviarlos correctamente formateados al origen de la petición.

Al igual que la función *index*, esta función también incluirá el decorador *@login_required* para evitar que ningún usuario no autenticado pueda llamarla.

El segundo elemento que interviene en el proceso de mostrar las visualizaciones al usuario son las *templates* o plantillas. Como se ha comentado anteriormente, estas plantillas están codificadas empleando el lenguaje de plantillas Mako, y, junto al conjunto de datos y contexto que ofrece la función *index* de *views.py*, generan la salida en forma de HTML que se muestra al usuario.

Las plantillas del módulo se pueden separar jerárquicamente en dos clases, por una parte están las plantillas que muestran cada una de las visualizaciones, y por otra una plantilla general que incluye todas las plantillas de visualizaciones y las organiza en pestañas.

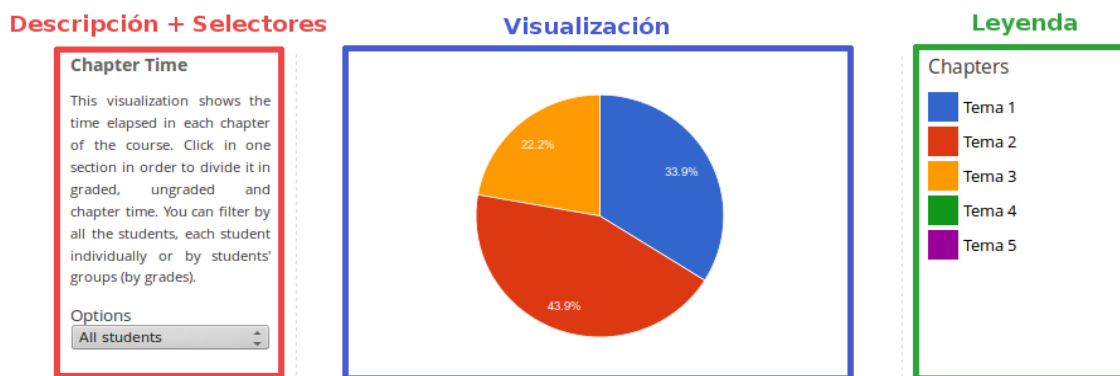


Figura 14: Divisiones de las visualizaciones

Todas las plantillas de visualizaciones emplean los mismos estilos CSS, por lo que su formato será muy similar. Como se observa en la Figura 14, estas plantillas tienen 3 bloques, o divs, diferenciados:

- **Descripción y selectores:** En el primer bloque se incluye una breve descripción de la visualización, así como los selectores requeridos en cada visualización. Esta descripción y selectores variaran dependiendo del usuario que hace la petición y su rol en el curso (estudiante o instructor). Los selectores serán manejados externamente por un fichero JavaScript.
- **Visualización:** En este bloque se muestra la visualización o gráfica. Esta visualización es dibujada y manejada por un fichero JavaScript externo.
- **Leyenda:** En el último bloque se muestra una leyenda para facilitar la lectura de la visualización. Al igual que en los otros dos bloques, el fichero JavaScript externo se encargará de hacer los cambios necesarios en esta leyenda cuando el usuario interactúe con la visualización.

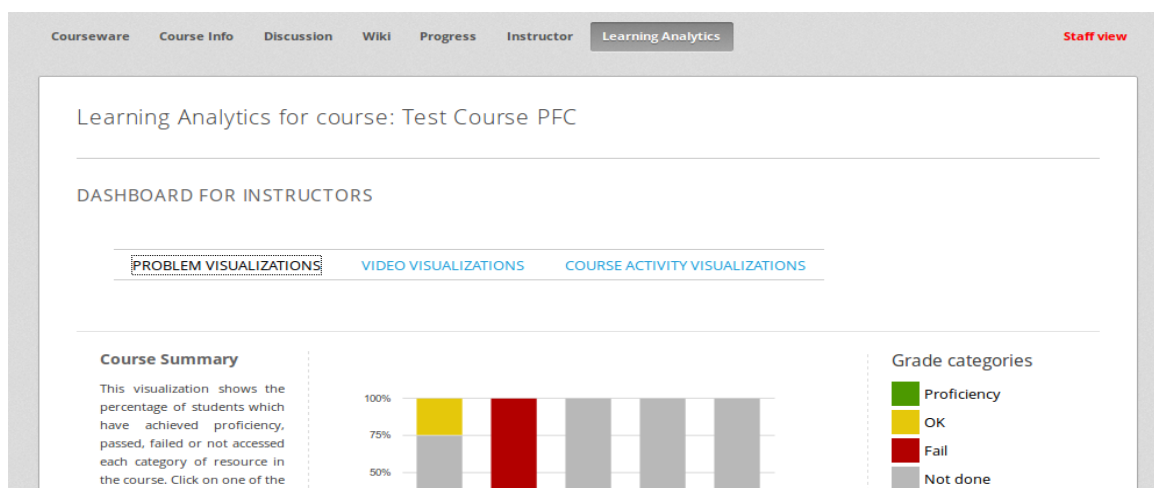


Figura 15: Visualización «Course Summary»

Por último, una plantilla general organizará cada una de las visualizaciones en distintas pestañas, según la clase de visualización que muestre (Figura 15). Por el momento las pestañas incluidas son: visualizaciones de vídeos, de problemas, y de actividad en el curso. Esta plantilla, también tendrá en cuenta el contexto que se le pase, para incluir o no incluir ciertas visualizaciones según el rol del usuario que haga la petición.

En el caso de las visualizaciones, se crearan incluyendo un JavaScript en el *div* indicado anteriormente. Como se comenta en el Capítulo 2, para dibujar estas gráficas, se emplea la librería *Google Visualizations* (o *Google Charts*). Los diseños y la creación de cada una de estas gráficas varían mucho entre ellas, por lo que se tratara cada una individualmente en el próximo capítulo.

Un elemento común a la mayoría de las visualizaciones, es que son capaces de realizar una petición AJAX al servidor que aloje la plataforma Open edX, cuando necesiten un conjunto de datos distinto al que tienen (por ejemplo, cuando se cambie el selector que indica el grupo de alumnos para el que se están mostrando los datos). Esta petición va dirigida a la función *update* del fichero *views.py* que se ha comentado anteriormente. Se incluye en la petición el ID del usuario que la realiza, el ID del curso, y un ID, para indicar que gráfica realiza la petición, cuyos valores están fijados en el fichero *views.py*. Una vez se recibe la respuesta, se actualizan los datos de la gráfica o se muestra un mensaje de error en caso de que la respuesta haya sido negativa.

3.2.4. Controlador: API de datos

El primer elemento de la parte controlador del módulo de *learning analytics* es una capa intermedia que obtenga los datos necesarios de la plataforma y los ofrezca al módulo de manera sencilla, ordenada y eficaz. Gracias al uso de este API, el mantenimiento del módulo se simplifica en gran manera, en caso de que la forma de obtención de algún dato cambie en un futuro, ya que funciona como capa de abstracción entre la plataforma Open edX y el módulo desarrollado en este proyecto.

El API de obtención de datos se encuentra codificado en el fichero *data.py* del módulo. Las funciones que ofrece este API van desde la obtención de módulos tales como cursos o

problemas, hasta la obtención de ciertos eventos almacenados en los *tracking logs* y se describen en detalle en el Capítulo 4 de esta memoria.

3.2.5. Controlador: Cálculo de estadísticas de alto nivel

El cálculo de las estadísticas de alto nivel para mostrar en las visualizaciones se realiza en el fichero *analytics.py*. Las funciones codificadas en este fichero se pueden separar en 4 tipos:

- Funciones encargadas de calcular las estadísticas de alto nivel de cada una de las visualizaciones
- Funciones auxiliares para el cálculo de estas estadísticas
- Funciones encargadas de almacenar en cada una de las tablas de la base de datos las estadísticas calculadas. El prefijo de estas funciones será *“update_DB_”*
- Funciones encargadas de leer de la base de datos las estadísticas calculadas, y formatearlas de manera que puedan ser interpretadas por las funciones de *views.py* cuando estas pidan los datos. El prefijo de estas funciones será *“get_DB_”*

Por cada una de las visualizaciones creadas existen, como mínimo, una función para almacenar los resultados de la estadística en la base de datos y otra para leer los datos y formatearlos para la capa de vista. Las funciones de cada una de las visualizaciones se describirán en el siguiente capítulo.

Además de estas funciones dedicadas a cada una de las visualizaciones, también existen dos funciones genéricas para varias de las visualizaciones, que son las encargadas de almacenar y leer de la base de datos la estructura del curso. En las *Tablas 4 Y 5* se describe su funcionamiento:

Tabla 4: Función *update_DB_course_struct*

	<i>update_DB_course_struct(course_key)</i>
Argumentos	• <i>course_key</i> : Localizador de curso (<i>CourseLocator</i>).
Devuelve	<i>None</i>
Descripción	La función recibe como argumento un localizador del curso del que se quiere almacenar la estructura. A partir de este localizador y haciendo uso de la función del API de datos <i>get_course_struct</i> se obtiene la estructura del curso y se almacena en la tabla creada a partir del modelo <i>CourseStruct</i> descrito anteriormente. Se crea una entrada por cada módulo de jerarquía superior (capítulo, <i>sequential</i> y <i>vertical</i>), que incluye el localizador de curso, el localizador, nombre, tipo, padre (si es que existe) y posición en la jerarquía del módulo, así como dos campos para indicar si el módulo está publicado y si es evaluable.

Tabla 5: Función `get_DB_course_struct`

	<code>get_DB_course_struct(course_key, include_verticals=False, include_unreleased=True)</code>
Argumentos	<ul style="list-style-type: none"> • <code>course_key</code>: Localizador de curso (<i>CourseLocator</i>). • <code>include_verticals</code>: <i>True</i> o <i>False</i> para que la salida de la función incluya o no los <i>verticals</i> del curso. Si no se indica nada, el valor por defecto es <i>False</i> • <code>include_unreleased</code>: <i>True</i> o <i>False</i> para que la salida de la función incluya o no las secciones no publicadas del curso. Si no se indica nada, el valor por defecto es <i>True</i>
Devuelve	<p>Un array con un diccionario por cada uno de los capítulos del curso pasado como parametro de la siguiente forma:</p> <pre>{ "id": Id de la entrada en la tabla CourseStruct (int), "module_id": Localizador del capítulo (BlockUsageLocator), "name": Nombre del capítulo (str), "graded": Indica si el capítulo es evaluable o no (bool), "type": String indicando que es un capítulo ("chapter") "sequentials": Array con diccionarios para cada sequential hijo }</pre> <p>Los diccionarios de cada <i>sequential</i> tienen la siguiente forma:</p> <pre>{ "id": Id de la entrada en la tabla CourseStruct (int), "module_id": Localizador del sequential (BlockUsageLocator), "name": Nombre del sequential (str), "graded": Indica si el sequential es evaluable o no (bool), "type": String indicando que es un sequential ("sequential"), "verticals": Array con diccionarios para cada vertical hijo }</pre> <p>Los diccionarios de cada <i>vertical</i> tienen la siguiente forma:</p> <pre>{ "id": Id de la entrada en la tabla CourseStruct (int), "module_id": Localizador del vertical (BlockUsageLocator), "name": Nombre del vertical (str), "graded": Indica si el vertical es evaluable o no (bool), "type": String indicando que es un vertical ("vertical") }</pre>
Descripción	<p>Dado un localizador de curso, la función devuelve un array con un diccionario por cada capítulo del curso que contiene los datos del capítulo almacenados en la tabla <i>CourseStruct</i>. Estos capítulos a su vez contendrán un array con un diccionario por cada <i>sequential</i> del que sean padre, también conteniendo los datos de dicho <i>sequential</i>. En caso de que el argumento <i>include_verticals</i> sea <i>True</i>, cada diccionario de <i>sequential</i> también contendrá un array con diccionario para cada <i>vertical</i> del que sea padre, conteniendo los datos de este <i>vertical</i>. Dependiendo del valor del argumento <i>include_unreleased</i> se incluirán o no los módulos que no estén aún publicados</p>

Las razones de emplear estas funciones, en lugar de recurrir a la función del API de datos de la que se obtienen primeramente la estructura del curso, *get_course_struct*, son por una parte, reducir la carga y tiempo de consulta, ya que esta función es lenta y pesada, al tener que cargar el módulo de curso y el de todos sus hijos para rellenar los datos, y por otra parte, ofrecer la posibilidad de realizar búsquedas complejas de la estructura, como obtener solo los capítulos u obtener los datos de un *sequential* en concreto.

3.2.6. Controlador: Tareas periódicas

El computo de las estadísticas para las visualizaciones, puede ser, en según que caso, muy pesado y lento. Calcular las estadísticas en tiempo real, cada vez que un usuario las solicite, se vuelve insostenible conforme los cursos crecen en tamaño y número de estudiantes. La solución a este problema, es calcular las estadísticas de forma periódica, y almacenar los resultados en las tablas de las bases de datos descritas anteriormente, de esta forma, cuando un usuario requiera los datos de una visualización, la carga en el servidor solamente será la de una consulta a la base de datos.

Para gestionar estos cálculos periódicos de las estadísticas de las visualizaciones, se empleará Celerybeat, un componente para organización de tareas periódicas, perteneciente al encolador de tareas Celery.

Puesto que la plataforma Open edX ya emplea Celery para varias tareas (envío masivo de emails, peticiones masivas de datos, o correcciones de problemas), el uso del componente Celerybeat no tiene mucha complejidad. A continuación se describen los pasos necesarios para la instalación y configuración del encolador de tareas periodicas Celerybeat:

- El primer paso es instalar y configurar el *broker* que Celery emplea por defecto, RabbitMQ. [29]
- El siguiente paso es definir la tarea o tareas que se quieren ejecutar periódicamente dentro del código del módulo. En el caso de el módulo de *learning analytics*, se ejecutara una tarea que llamara a todas las funciones “*update_DB_**” de *analytics.py* para cada uno de los cursos existentes en la plataforma. La función se encontrara en el fichero *tasks.py* y su nombre es *update_DB_analytics*. Para indicar a Celery que esta función es una tarea que debe gestionar basta con añadir el decorador *@task* a dicha función.
- Una vez el *broker* RabbitMQ este en funcionamiento y la tarea que queremos ejecutar periódicamente este correctamente definida, habrá que inicializar en el archivo de configuración del entorno que vamos a emplear, la variable global CELERYBEAT_SCHEDULE de la siguiente forma:

```
CELERYBEAT_SCHEDULE = {
    "[Nombre de tarea]": {
        "task": [Ruta de la tarea],
        "schedule": [Periodo de la tarea]
    }
}
```


La ruta de la tarea en nuestro caso será *'learning_analytics.tasks.update_DB_analytics'*. El periodo de la tarea se incluye como un *timedelta* indicando cada cuanto tiempo se debe ejecutar la tarea (por ejemplo, si se quiere ejecutar cada 30 minutos, el *timedelta* será *timedelta(minutes=30)*)

Por último, hay que arrancar Celery mediante *paver*, indicando cual es el entorno de configuración que se empleara (*devstack_analytics* en este caso).

Gracias al uso de Celerybeat para el calculo de las estadísticas, la carga en el servidor se rebajara drásticamente, aunque será necesario estudiar el tamaño de los cursos y número de estudiantes en la plataforma, así como la potencia del servidor en el que se aloje la plataforma, para determinar cual es el periodo óptimo de la llamada a la tarea para cada caso.



CAPÍTULO 4

4. API DE DATOS

Como se ha comentado anteriormente, uno de los principales objetivos de este proyecto, es la creación de un API para obtener, de manera sencilla, datos que almacena la plataforma Open edX. Esta plataforma, almacena una gran cantidad de datos, pero la obtención de estos no siempre es sencilla, ya que están pensados para objetivos concretos, generalmente para el funcionamiento con módulos mediante xModule o xBlocks.

Por tanto, el principal objetivo de este API es proporcionar al módulo de *learning analytics* los datos que necesite de manera sencilla, ordenada y eficaz.

Otra ventaja que se obtiene al emplear esta capa intermedia, es la separación del módulo de la obtención de datos de la plataforma, que, debido al continuo desarrollo de la plataforma Open edX, están en constante cambio. Gracias a esta abstracción, el mantenimiento del módulo se simplifica en gran manera, ya que, en caso de que la forma de obtención de algún dato cambie en un futuro, bastara con corregir la forma de obtener los datos en el API de datos, para que el resto del módulo siga funcionando de igual manera.

Por último, esta capa intermedia también está diseñada para que pueda ser empleada por cualquier desarrollador que quiera bien, extender el módulo, o emplear estos datos para su propio desarrollo. Debido a esto, no se han incluido en este API muchas funciones de obtención de datos muy específicas a las visualizaciones que se desarrollaran para el módulo, las cuales se incluirán en el fichero *analytics.py* y se describen en el siguiente capítulo.

Dado que este API tiene como objetivo que sea empleado y ampliado por la comunidad de desarrolladores de Open edX, como punto de partida solo se incluirán funciones que vayan a ser usadas por el módulo de *learning analytics* que se está desarrollando. Posiblemente con este planteamiento no se cubran todos los datos que puede ofrecer la plataforma, pero ya que el desarrollo de este módulo es de código abierto, se espera que el API sea ampliado por el resto de la comunidad para obtener otros datos que puedan resultar útiles fuera de este proyecto.

El API de datos, al igual que el resto del módulo, forma parte de un proyecto común en el que están trabajando o han trabajado un equipo de varias personas. A continuación se describen las funciones que se han incluido en el API en este proyecto, describiendo su funcionalidad, así como la forma correcta de emplearlas. Junto a estas funciones habrá otras tantas en el API, desarrolladas por el resto del equipo que trabaja en el módulo, pero que no son relevantes para este proyecto. Las funciones incluidas se dividirán según el tipo de datos que ofrezcan en: datos de cursos, de problemas, de eventos y de localizaciones keys o URLs.

4.1. CURSOS

En las *Tablas 6-13* se describe la funcionalidad y modo de uso de las funciones incluidas en el API, que ofrecen datos o herramientas relacionadas con los cursos de la plataforma.

Tabla 6: Función del API *get_courses_list*

	<i>get_courses_list()</i>
Argumentos	Ninguno
Devuelve	Array de módulos descriptores de cursos (<i>[CourseDescriptorWithMixins, ...]</i>)
Descripción	Devuelve una lista con los descriptores de los módulos de todos los cursos que existan en la plataforma

Tabla 7: Función del API *get_course_key*

	<i>get_course_key(course_id)</i>
Argumentos	· <i>course_id</i> : String con la ruta de un curso en la forma antigua (“[ORG]/[COURSE]/[RUN]”)
Devuelve	Localizador de curso (<i>CourseLocator</i>)
Descripción	Indicando el curso con la antigua sintaxis, devuelve el localizador del curso con el nuevo estándar, <i>OpaqueKeys</i>

Tabla 8: Función del API *get_course_module*

	<i>get_course_module(course_key)</i>
Argumentos	· <i>course_key</i> : Localizador de curso (<i>CourseLocator</i>)
Devuelve	Descriptor de curso (<i>CourseDescriptorWithMixins</i>)
Descripción	Devuelve el módulo del curso indicado en el localizador de curso

Tabla 9: Función del API *get_course_struct*

	<i>get_course_struct(course)</i>
Argumentos	· <i>course</i> : Descriptor de curso (<i>CourseDescriptorWithMixins</i>)
Devuelve	Diccionario con la estructura del curso, incluyendo los módulos Course, Chapters, Sequentials y Verticals. La forma del diccionario es la siguiente: { “id”: Localizador del curso (<i>CourseLocator</i>), “name”: Nombre del curso (<i>str</i>), “chapters”: Array con un diccionario para cada capítulo del curso [{ “id”: Localizador del capítulo (<i>BlockUsageLocator</i>), “name”: Nombre del capítulo (<i>str</i>), “released”: Indica si el capítulo esta publicado (<i>bool</i>), “sequentials”: Array con un diccionario para cada <i>sequential</i> [{ “id”: Localizador del <i>sequential</i> (<i>BlockUsageLocator</i>), “name”: Nombre del <i>sequential</i> (<i>str</i>), “graded”: Indica si el <i>sequential</i> es evaluable (<i>bool</i>), “released”: Indica si el secuencial esta publicado (<i>bool</i>), “verticals”: Array con un diccionario para cada <i>vertical</i> [{ “id”: Localizador del <i>vertical</i> (<i>BlockUsageLocator</i>), “name”: Nombre del <i>vertical</i> (<i>str</i>), “graded”: Indica si el <i>vertical</i> es evaluable(<i>bool</i>),

	<i>get_course_struct(course)</i> “released”: Indica si el vertical esta publicado (<i>bool</i>) }, {...}, ...] }, {...}, ...] }, {...}, ...] }
Descripción	Pasando como argumento el descriptor del curso, devuelve un diccionario con el contenido de los módulos de más alta jerarquía (curso, capítulos, sequentials y verticals) del curso dado.

Tabla 10: Función del API *get_course_grade_cutoff*

	<i>get_course_grade_cutoff(course)</i>
Argumentos	• course: Descriptor de curso (<i>CourseDescriptorWithMixins</i>)
Devuelve	Valor numérico de la nota de corte del curso (<i>float</i>)
Descripción	Devuelve el valor numérico de la nota de corte del curso pasado como argumento

Tabla 11: Función del API *get_course_students*

	<i>get_course_students(course_key)</i>
Argumentos	• course_key: Localizador de curso (<i>CourseLocator</i>)
Devuelve	Array de elementos de la clase <i>User</i> de Django.auth (<i>[User, ...]</i>)
Descripción	Devuelve una lista de los usuarios del curso indicado en el localizador de curso pasado como argumento

Tabla 12: Función del API *dump_full_grading_context*

	<i>dump_full_grading_context(course)</i>
Argumentos	• course: Descriptor de curso (<i>CourseDescriptorWithMixins</i>)
Devuelve	Diccionario con dos elementos: - “ <i>graded_sections</i> ”: Array con cada un diccionario por cada una de las secciones evaluables del curso de la siguiente forma: { “ <i>category</i> ”: Categoría de la sección (<i>str</i>), “ <i>label</i> ”: Etiqueta de la sección (<i>str</i>), “ <i>released</i> ”: Indica si la sección esta publicada (<i>bool</i>), “ <i>name</i> ”: Nombre de la sección (<i>str</i>), “ <i>problems</i> ”: Array con los descriptors de cada uno de los problemas de la sección (<i>[CapaDescriptorWithMixins, ...]</i>), nota máxima de la sección (<i>float</i>) } - “ <i>weight_subsections</i> ”: Array con un diccionario por cada una de las distintas categorías posibles de las secciones de la siguiente forma: { “ <i>category</i> ”: Nombre de la categoría (<i>str</i>), “ <i>type</i> ”: Tipo del evaluador de la categoría (<i>str</i>), “ <i>weight</i> ”: Peso de la categoría sobre el total de la nota (<i>float</i>) }
Descripción	Dado un descriptor de curso, devuelve un diccionario con un array con cada una de las secciones evaluables del curso, y otro array con cada una de las posibles categorías de esas secciones y el peso total de estas sobre la nota final

Tabla 13: Función del API *get_course_blocks*

	<i>get_course_blocks(course)</i>
Argumentos	<ul style="list-style-type: none"> • <i>course</i>: Descriptor de curso (<i>CourseDescriptorWithMixins</i>)
Devuelve	Un diccionario con las Ids de los bloques (hijos de <i>verticals</i>) del curso como palabras clave, teniendo cada entrada el siguiente contenido: <pre>{[ID DE BLOQUE]: { "chapter": Localizador de capítulo (BlockUsageLocator), "sequential": Localizador de <i>sequential</i> (BlockUsageLocator), "block": Localizador de bloque (BlockUsageLocator) } [ID DE BLOQUE]: }</pre>
Descripción	Dado un descriptor de curso, devuelve un diccionario con cada uno de los bloques hijos del curso. La palabra clave de cada entrada del diccionario será la Id del bloque, y el contenido será el localizador del bloque y el de sus padres.

4.2. PROBLEMAS

En las *Tablas de 14-17* se describen las funciones del API relacionadas con los problemas del curso.

Tabla 14: Función del API *is_problem_done*

	<i>is_problem_done(course_key, user, problem_descriptor)</i>
Argumentos	<ul style="list-style-type: none"> • <i>course_key</i>: Localizador de curso (<i>CourseLocator</i>) • <i>user</i>: Usuario que se quiere consultar (<i>User</i>) • <i>problem_descriptor</i>: Descriptor del problema (<i>CapaDescriptorWithMixins</i>)
Devuelve	<i>True</i> o <i>False</i>
Descripción	Dado un localizador de curso, un usuario, y un descriptor de problema, indica si dicho usuario ha realizado o no el problema dado

Tabla 15: Función del API *section_max_grade*

	<i>section_max_grade(course_key, problem_descriptors)</i>
Argumentos	<ul style="list-style-type: none"> • <i>course_key</i>: Localizador de curso (<i>CourseLocator</i>) • <i>problem_descriptors</i>: Array con los descriptores de todos los problemas evaluables de la sección (<i>[CapaDescriptorWithMixins, ...]</i>)
Devuelve	Valor numérico de la nota máxima que se puede obtener en la sección indicada (<i>Float</i>)
Descripción	Dado un localizador de curso y los descriptores de los problemas de una sección, devuelve la suma de la nota máxima de estos problemas, es decir, la nota máxima que un alumno puede obtener en esa sección

Tabla 16: Función del API *get_problem_score*

	<i>get_problem_score(course_key, user, problem_descriptor)</i>
Argumentos	<ul style="list-style-type: none"> • <i>course_key</i>: Localizador de curso (<i>CourseLocator</i>) • <i>user</i>: Usuario que se quiere consultar (<i>User</i>) • <i>problem_descriptor</i>: Descriptor del problema (<i>CapaDescriptorWithMixins</i>)
Devuelve	Un par de valores numéricos con la nota del usuario y la nota máxima del problema (<i>(float, float)</i>)
Descripción	Dado un localizador de curso, un usuario, y un descriptor de problema, devuelve la nota del usuario en el problema dado, así como la nota máxima del problema

Tabla 17: Función del API *is_problem_for_course*

	<i>is_problem_for_course(problem_id, course_key)</i>
Argumentos	<ul style="list-style-type: none"> • <i>problem_id</i>: Id del problema a consultar (<i>str</i>) • <i>course_key</i>: Localizador de curso (<i>CourseLocator</i>)
Devuelve	<i>True</i> o <i>False</i> dependiendo de si el problema pertenece o no al curso dado
Descripción	Dado un localizador de curso y un identificador de problema, devuelve <i>True</i> o <i>False</i> dependiendo de si el problema pertenece al curso dado o no

4.3. EVENTOS

En las *Tablas 18, 19 y 20* se describen las funciones del API encargadas de obtener datos de eventos almacenados en los *Tracking Logs*. Note que estas funciones solo funcionaran correctamente si el almacenamiento de *Tracking Logs* en la base de datos SQL esta activado en la plataforma.

Tabla 18: Función del API *get_course_events_sql*

	<i>get_course_events_sql(course_key, student)</i>
Argumentos	<ul style="list-style-type: none"> • <i>course_key</i>: Localizador de curso (<i>CourseLocator</i>) • <i>student</i>: Usuario que se quiere consultar (<i>User</i>) o Nombre de usuario (<i>String</i>)
Devuelve	Una lista de objetos <i>TrackingLog</i> con los eventos generados por el usuario dado (<i>[TrackingLog, ...]</i>)
Descripción	<p>Dado un estudiante y un curso, devuelve una lista filtrada de todos los eventos de cierto tipo para el curso dado. La lista se devuelve ordenada por orden cronológico. Nótese que para que esta función sea valida, el almacenamiento de los <i>tracking logs</i> en la base de datos SQL debe estar activado. Los tipos de eventos que filtra esta función son los siguientes:</p> <ul style="list-style-type: none"> - Eventos generados en el navegador: Los que sean del tipo “<i>seq_goto</i>”, “<i>seq_next</i>”, “<i>page_dose</i>”, “<i>show_transcript</i>” o “<i>load_video</i>” - Eventos generados en el servidor: Todos los que se generen para el curso dado

Tabla 19: Función del API *get_course_access_events_sql*

	<i>get_course_access_events_sql(course_key, student)</i>
Argumentos	<ul style="list-style-type: none"> • <i>course_key</i>: Localizador de curso (<i>CourseLocator</i>) • <i>student</i>: Usuario que se quiere consultar (<i>User</i>) o Nombre de usuario (<i>str</i>)
Devuelve	Una lista de objetos <i>TrackingLog</i> con los eventos de acceso a las secciones del curso generados por el usuario dado (<i>[TrackingLog, ...]</i>)
Descripción	<p>Dado un estudiante y un curso, devuelve una lista filtrada de todos los eventos de acceso a las secciones del curso, para el curso dado. La lista se devuelve ordenada por orden cronológico. Nótese que para que esta función sea valida, el almacenamiento de los <i>tracking logs</i> en la base de datos SQL debe estar activado. Los tipos de eventos que filtra esta función son los siguientes:</p> <ul style="list-style-type: none"> - Eventos generados en el navegador: Los que sean del tipo “<i>seq_prev</i>”, “<i>seq_goto</i>” o “<i>seq_next</i>” - Eventos generados en el servidor: Todos los eventos que tengan accesos a secciones del curso

Tabla 20: Función del API *get_problem_history_sql*

	<i>get_problem_history_sql(course_key, student)</i>
Argumentos	<ul style="list-style-type: none"> • <i>course_key</i>: Localizador de curso (<i>CourseLocator</i>) • <i>student</i>: Usuario que se quiere consultar (<i>User</i>)
Devuelve	Una lista de objetos <i>TrackingLog</i> con los eventos de problemas contestados generados por el usuario dado (<i>[TrackingLog, ...]</i>)
Descripción	<p>Dado un estudiante y un curso, devuelve una lista filtrada de todos los eventos de contestaciones de problemas del curso, para el curso y el alumno dado. La lista se devuelve ordenada por orden cronológico. Nótese que para que esta función sea valida, el almacenamiento de los <i>tracking logs</i> en la base de datos SQL debe estar activado. Los tipos de eventos que filtra esta función son todos los eventos “<i>problem_check</i>” y “<i>problem_rescore</i>” para el usuario y el curso dado</p>

4.4. LOCALIZADORES, KEYS Y URLS

Debido al desarrollo constante de la plataforma Open edX, nos encontramos con que actualmente se emplean distintas formas de referirse a los módulos del curso (problemas, *sequentials*, cursos, etc). En los inicios de la plataforma se empleaban localizadores en forma de URL o String. Con el tiempo se incluyeron objetos de tipo *Location* para referirse a ellos y añadir funcionalidades extra. El último cambio ha sido crear un nuevo tipo de objetos, llamados *Opaque Keys* para substituir a los objetos *Location*.

Debido a esto, en la plataforma coexisten los tres tipos de identificadores de módulos, aunque se esta tendiendo a eliminar los métodos antiguos. Para evitar problemas con la forma de identificar a los modulos, se han incluido en el API varias herramientas para

comparar, obtener y tratar los módulos de forma que se abstraiga el tipo de localizador empleado. En las *Tablas 21-24* se describen estas funciones.

Tabla 21: Función del API *is_same_course*

	<i>is_same_course(course_1, course_2)</i>
Argumentos	• <i>course_1</i> y <i>course_2</i> : Localizadores o identificadores de curso en cualquier formato soportado: <i>CourseLocator</i> , <i>String</i> , <i>String</i> conteniendo la ruta del curso, etc
Devuelve	<i>True</i> o <i>False</i> dependiendo de si es el mismo curso o no
Descripción	Compara dos cursos dados en dos formatos distintos (Localizadores antiguos, localizadores de <i>OpaqueKeys</i> , <i>Strings</i> conteniendo la ruta del curso, etc) y devuelve <i>True</i> o <i>False</i> dependiendo de si apuntan al mismo curso o no

Tabla 22: Función del API *get_course_from_url*

	<i>get_course_from_url(url)</i>
Argumentos	• <i>url</i> : Identificador de curso en formato <i>String</i> en cualquiera de las formas soportadas (identificador antiguo, <i>String</i> con la ruta del curso incluida, etc)
Devuelve	Localizador de curso (<i>CourseLocator</i>) o <i>None</i> en caso de que no exista el curso
Descripción	Dada una URL con la ruta del curso contenida en ella en cualquiera de los formatos soportados, devuelve el localizador de curso si encuentra alguno que coincida con la URL dada

Tabla 23: Función del API *get_locations_from_url*

	<i>get_locations_from_url(url, course_blocks=None)</i>
Argumentos	• <i>url</i> : URL del bloque de la forma “.../courses/[ORG]/[COURSE]/[RUN]/[Chapter ID]/[Sequential ID]/...” • <i>course_blocks</i> : Diccionario de la forma de la salida de la función <i>get_course_blocks</i> . Si no se pasa nada en este campo, se llamará a dicha función para obtener el diccionario.
Devuelve	Tres valores con los localizadores de curso, capítulo y secuencial, si existen en la URL pasada, o <i>None</i> si cualquiera de los 3 no existen ((<i>CourseLocator</i> , <i>BlockUsageLocator</i> , <i>BlockUsageLocator</i>))
Descripción	Devuelve los localizadores del curso, capítulo y secuencial de la URL dada. Si alguno de los 3 campos no existe en la URL, devuelve <i>None</i> en el lugar de ese campo.

Tabla 24: Función del API *compare_locations*

	<i>compare_locations(loc1, loc2, course_key=None)</i>
Argumentos	• <i>loc1</i> y <i>loc2</i> : Localizador de algún módulo del curso en cualquiera de los siguientes formatos: localizador antiguo (<i>Location</i>), localizador nuevo (<i>BlockUsageLocator</i>), o url del módulo (<i>str</i> o <i>unicode</i>) • <i>course_key</i> : Localizador de curso (<i>CourseLocator</i>). En caso de no incluir

	este campo se obtendrá de alguno de los dos localizadores si es posible (si son formato <i>Location</i> o <i>BlockUsageLocator</i>)
Devuelve	<i>True</i> o <i>False</i> dependiendo de si las dos localizaciones pasadas como parámetro se refieren al mismo módulo o no
Descripción	Pasando dos localizadores de módulo como parámetro en distintos posibles formatos, devuelve <i>True</i> o <i>False</i> dependiendo de si se refieren al mismo módulo o no



CAPÍTULO 5

5. VISUALIZACIONES

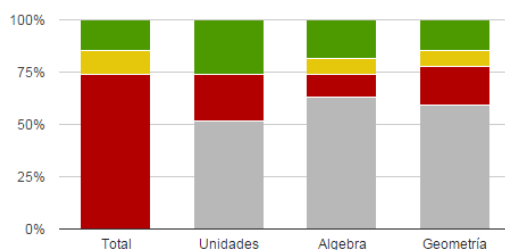
En este capítulo se describen cada una de las visualizaciones que se incluirán en el módulo de *learning analytics*. Se explica la motivación para la inclusión de dichas visualizaciones en el módulo, así como las aportaciones y utilidad que tienen cada una para la mejora del proceso educativo. También se explica la forma correcta de emplearlas y las posibilidades de interacción que tienen. Por último, se describe su funcionamiento interno y las tablas de la base de datos que emplean cada una.

5.1. COURSE SUMMARY

La primera visualización que se trata en este capítulo, pertenece a la categoría de visualizaciones de problemas y muestra un diagrama de barras apiladas con cada una de las categorías evaluables del curso (*homework*, laboratorios, exámenes, etc) (Figura 16). En cada una de estas barras se muestra, diferenciados por el color, el porcentaje de alumnos que ha sacado una nota excepcional, ha aprobado, ha suspendido, o no ha hecho los ejercicios evaluables de la categoría. La gráfica es interactiva, y se puede consultar el desglose de apartados de cada una de las categorías.

Course Summary

This visualization shows the percentage of students which have achieved proficiency, passed, failed or not accessed each category of resource in the course. Click on one of the bars in order to see the results decomposed in each item within the category.



Grade categories



Figura 16: Visualización "Course Summary"

Debido a que en este modulo tiene como política no mostrar datos de otros usuarios a los estudiantes, esta visualización solo estará disponible para los instructores del curso, y el staff de la plataforma.

5.1.1. Motivaciones y utilidad

Hasta el momento, la plataforma Open edX, no ofrecía a los instructores de los cursos ninguna forma rápida y sencilla de obtener una idea general de como esta funcionando el

curso. Si querían hacerse una idea de esto, tenían que observar las calificaciones de las categorías una a una, lo cual en un curso masivo, con un gran número de alumnos, es prácticamente imposible.

Con esta visualización, en un rápido vistazo, los instructores pueden ver, en primer lugar, si el nivel de los distintos tipos de categorías esta igualado (o esta como pretendían que fuera, por ejemplo, los exámenes con algo más de dificultad que los ejercicios de los capítulos de teoría). Por otra parte, pueden localizar secciones del curso que estén causando problemas a los alumnos, o que sean demasiado fáciles, pudiendo así modificar esos apartados para futuros lanzamientos del curso.

Hay que tener en cuenta, que esta visualización, solo tiene en cuenta las calificaciones obtenidas, y no otros elementos como el tiempo dedicado, las veces que se han repetido, etc. Para estos elementos, se incluyen otras visualizaciones que el instructor también podrá consultar.

5.1.2. Modo de empleo

El uso de esta visualización es sencillo, ya que al ser una gráfica general para todo el curso, no incluirá selectores para seleccionar alumnos o grupos de alumnos. La primera vez que se cargue la gráfica se muestra, como se ha comentado anteriormente, las distintas categorías que tenga el curso. Si el instructor quiere ver el desglose de alguna categoría, le basta con pulsar en ella y se mostrarán todos los apartados de dicha categoría que hayan sido publicados hasta el momento. Si quiere volver a la vista general de categorías, tendrá que pulsar en cualquiera de los apartados.

5.1.3. Funcionamiento interno

En esta visualización, el grueso del código se encuentra en la función, *sort_homework* del fichero *analytics.py*. Aparte de esta función están la función *update_DB_course_homework*, que hará uso de la anterior para obtener los datos y los almacenará en el modelo de la base de datos que se describe a continuación, y la función *get_DB_course_homework*, que devolverá los datos de la base de datos de forma que sea legible por las funciones de *views.py*.

El funcionamiento de la función *sort_homework* es sencillo, simplemente obtiene los datos de notas de los alumnos en cada sección de la tabla *StudentGrades* descrita en el capítulo anterior, y rellena un diccionario con cada una de las categorías y sus secciones, indicando el número de alumnos que han sacado nota excepcional, aprobado, suspendido o no han hecho los ejercicios de la sección. El cálculo del límite a partir del cual una nota se considera excepcional se calcula de la siguiente forma, en función del límite de aprobado del curso:

$$(1 - \text{pass_limit}) / 2 + \text{pass_limit}$$

En la Tabla 25 se muestran los campos de la tabla de la base de datos creada a partir del modelo *SortGrades*, que se emplea para almacenar los datos calculados en *sort_homework*.

Tabla 25: Tabla de la base de datos SortGrades

Columna	Tipo	Null	Key	Descripción
<i>id</i>	Int	NO	PRI	ID de la entrada en la tabla
<i>course_id</i>	CourseKeyField	NO	MUL	ID del curso. Este campo junto con <i>label</i> y <i>sort_type</i> tienen que ser únicos
<i>sort_type</i>	Char	NO		Tipo de entrada, entre dos posibles valores: Graded Section (GS) o Weight Section (WS). Este campo junto con <i>label</i> y <i>course_id</i> tienen que ser únicos
<i>category</i>	Char	NO		Nombre de la categoría
<i>label</i>	Char	NO		Etiqueta de la categoría
<i>name</i>	Char	NO		Nombre de la sección o categoría
<i>num_not</i>	Int	NO		Número de estudiantes que no han hecho los ejercicios de la sección o categoría
<i>num_fail</i>	Int	NO		Número de estudiantes que no han suspendido los ejercicios de la sección o categoría
<i>num_pass</i>	Int	NO		Número de estudiantes que han aprobado los ejercicios de la sección o categoría
<i>num_prof</i>	Int	NO		Número de estudiantes que han sacado una nota excepcional los ejercicios de la sección o categoría
<i>last_calc</i>	DateTime	NO		Fecha de la última modificación de la entrada

5.2. STUDENT GRADES

En esta visualización, perteneciente, al igual que en la anterior, a la categoría de visualizaciones de problemas, muestra un diagrama de barras con todas las categorías evaluables del curso (*homework*, laboratorios, exámenes, etc) y el total de todas las categorías (*Figura 17*). Por cada una de estas categorías se mostrará la nota media obtenida en ellas por los alumnos, además la barra se mostrará de distintos colores según la nota (nota excepcional, aprobado o suspenso). La gráfica es interactiva, y se podrá obtener un desglose de cada una de las categorías.

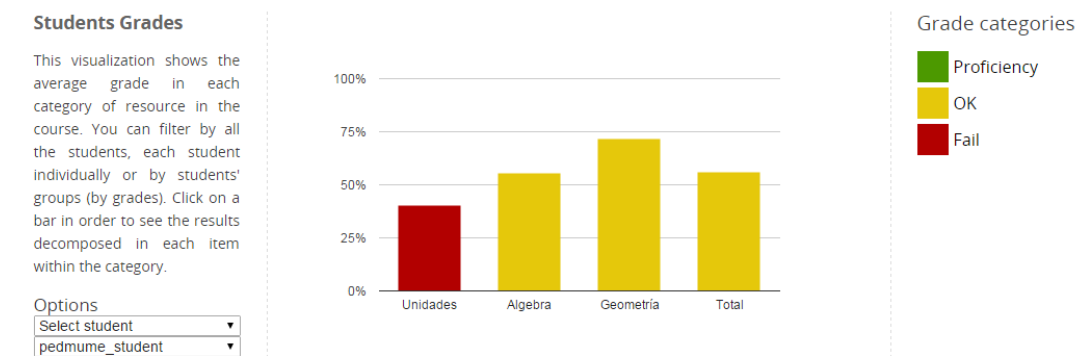


Figura 17: Visualización "Student Grades"

En este caso, si que es interesante para los alumnos el acceso a esta gráfica, por lo que podrán acceder a ella, pudiendo consultar únicamente sus propias calificaciones. También se incluirá un selector para el instructor para que seleccione las notas de un usuario concreto o la media de un grupo de alumnos.

5.2.1. Motivaciones y utilidad

Al igual que con la anterior visualización, el instructor podrá tener rápidamente una idea general de como están funcionando las categorías evaluables del curso. Aunque la información que ofrece al instructor, quizá no sea tan útil como la anterior visualización, ya que una nota media puede verse contaminada por varios factores (alumnos que no hagan el ejercicio, por ejemplo), pero sirve al instructor para analizar con algo más de profundidad ciertas secciones o categorías.

La opción de mostrar la nota media de grupos de alumnos (alumnos con el curso suspendido, aprobados o con nota excepcional) puede ser muy útil para los instructores, ya que no es lo mismo que la nota media de los alumnos suspendidos sea baja a que este rozando el aprobado. Otro uso muy útil de esta visualización es consultar las calificaciones de alumnos concretos por parte de los profesores.

Por último, esta visualización será de gran utilidad para los alumnos, que podrán consultar sus notas en todas las secciones de forma rápida y sencilla.

5.2.2. Modo de empleo

El modo de empleo de esta visualización es muy similar al de la anterior. Muestra una barra por cada una de las categorías evaluables del curso y una media computada de todas las categorías, indicando la nota media de todos los alumnos del curso en dichas categorías. Las barras se muestran de distinto color en función de la nota media (suspense, aprobado, o nota excepcional).

La gráfica es interactiva: si se pulsa sobre cualquiera de las categorías, se mostrara un desglose con todas las secciones que componen dicha categoría. Para volver a mostrar todas las categorías, basta con pulsar sobre la barra que indica el total de la categoría seleccionada.

Esta visualización también incluirá un selector que permitirá a los instructores seleccionar que la gráfica se forme con los datos de todos los alumnos, de algún grupo de alumnos en concreto (aprobados, suspensos o con nota excepcional) o seleccionar un alumno concreto.

5.2.3. Funcionamiento interno

Las funciones que engloban la funcionalidad del cálculo de la estadística de esta visualización son las siguientes:

- La función `get_student_grades_course_struct`, devuelve un diccionario con todas las secciones y categorías evaluables del curso.
- La función `get_student_grades`, que recibe dicho diccionario como parámetro, recorre todos los problemas de las categorías del curso, obteniendo la nota del alumno pasado como parámetro. Para calcular dichas notas hace uso de la función `get_problem_score` del API de datos tratado en el capítulo anterior. Por último se encarga de obtener la nota total de cada categoría sumando los resultados obtenidos en cada una de sus secciones.
- La función encargada de almacenar los resultados en la base de datos, `update_DB_student_grades`, en este caso también se encarga de, una vez obtenidos los datos para todos los estudiantes, calcular la media de notas para cada uno de los grupos de usuarios, y para el total de todos los usuarios registrados. Para el cálculo de estas medias se ayuda de las funciones auxiliares `add_student_grades` y `mean_student_grades`.

El proceso del cálculo de la estadística es el siguiente: la función `update_DB_student_grades` obtiene el diccionario llamando a `get_student_grades_course_struct`, y lo rellena, pasándoselo a la función `get_student_grades` para cada uno de los alumnos inscritos en el curso. Una vez tenga los datos de todos los estudiantes, calcula la media para los grupos de estudiantes y para la estadística con el total de los estudiantes, y almacena todos estos datos en la tabla de la base de datos creada a partir del modelo `StudentGrades`.

En la Tabla 26 se muestran los campos de la tabla de la base de datos en la que se almacenan los resultados, creada a partir del modelo `StudentGrades`.

Tabla 26: Tabla de la base de datos `StudentGrades`

Columna	Tipo	Null	Key	Descripción
<code>id</code>	int	NO	PRI	ID de la entrada en la tabla
<code>student_id</code>	int	NO		ID del estudiante. Este campo, junto con <code>course_id</code> tienen que ser únicos. Habrá 4 constantes negativas que se emplearán como identificadores de grupos de alumnos: -1: Todos los estudiantes -2: Estudiantes con nota excepcional -3: Estudiantes aprobados -4: Estudiantes suspensos
<code>course_id</code>	CourseKey	NO	MUL	ID del curso. Este campo, junto

Columna	Tipo	Null	Key	Descripción
				con <i>student_id</i> tienen que ser únicos.
<i>grades</i>	Text	NO		<p>Diccionario con las notas del alumno para cada sección. El diccionario tiene dos claves que incluyen un array de elementos.</p> <p>La primera clave, “<i>graded_sections</i>”, incluye un array con un elemento para cada una de las secciones evaluables del curso de la siguiente forma:</p> <pre>{ “category”: Categoría de la sección, “label”: Etiqueta de la sección, “name”: Nombre de la sección, “total”: Nota máxima de la sección, “score”: Nota en la categoría, “done”: Indica si el usuario ha hecho o no algún ejercicio de la sección }</pre> <p>La segunda clave, “<i>weight_subsections</i>”, incluye un array con un elemento para cada una de las categorías de la siguiente forma:</p> <pre>{ “category”: Categoría, “weight”: Peso de la nota de la categoría sobre el total, “total”: Nota máxima de la categoría, “score”: Nota en la categoría, “done”: Indica si el usuario ha hecho o no algún ejercicio de la categoría }</pre>
<i>grade_group</i>	Char	NO		Grupo de alumnos en el que está el alumno según su nota media. Los valores posibles son: “PROF”, “OK” y “FAIL”
<i>last_calc</i>	DateTime	NO		Fecha de la última modificación de la entrada

Esta tabla, como se explico en el Capítulo 3, se empleara también por muchas de las otras visualizaciones, en concreto las que muestren selectores para seleccionar datos por grupos de estudiantes, para consultar en que grupo de estudiantes está cada uno de los usuarios.

5.3. CHAPTER TIME

Esta visualización, pertenece a la categoría de *visualizaciones de actividad en el curso*. Muestra un diagrama de tarta en el que se muestra el tiempo dedicado por los alumnos a cada una de los capítulos publicados del curso (*Figura 18*). El gráfico es interactivo, y permite ver un desglose de cada uno de los capítulos mostrando el tiempo dedicado al contenido no evaluable (o de teoría), evaluable (o ejercicios) y a la introducción del capítulo.

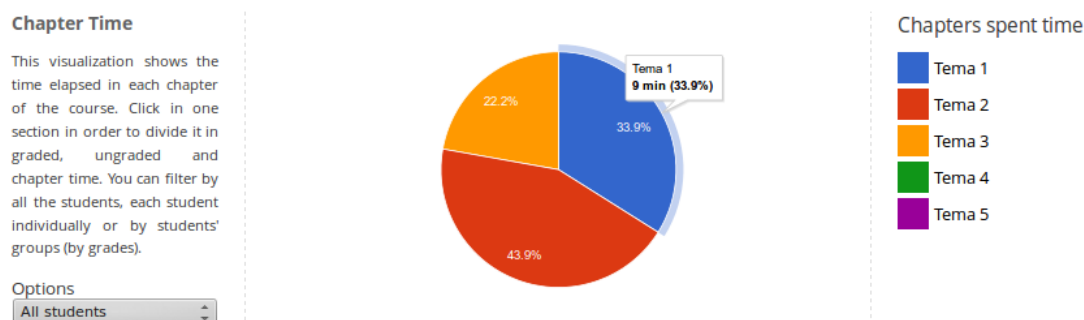


Figura 18: Visualización "Chapter Times"

Esta visualización puede ser consultada tanto por instructores como por alumnos (pudiendo consultar solo sus propios datos), e incluye un selector para los instructores para seleccionar alumnos o grupos de alumnos.

5.3.1. Motivaciones y utilidad

En muchas ocasiones, los profesores de un curso, pueden estimar un tiempo dedicado por los alumnos a un capítulo, o un apartado de teoría, que acabe resultando menor o mayor en la realidad. Esto puede deberse a diversos motivos: un mal cálculo de la cantidad de contenido de la sección, que dicha sección tenga una dificultad extra, o que sea más sencilla de lo normal, ya sea por la naturaleza de la materia, o por la forma en que esta explicada.

Gracias a esta visualización, los instructores podrán detectar rápidamente si los alumnos están dedicando más o menos tiempo del estimado a alguno de los capítulos. Al pulsar sobre estos capítulos también se muestra cuanto de este tiempo esta dedicado a contenido no evaluable (o de teoría) y a tiempo evaluable.

Una vez detectadas estas diferencias con respecto al tiempo estimado, los instructores podrán actuar, ya sea cambiando la forma de explicar o acortando/alargando el contenido para futuros lanzamientos del curso, o reforzar contenidos que estén costando más tiempo a los alumnos.

Por otra parte los alumnos también pueden consultar sus tiempos, viendo cuales de los capítulos les esta costando más o menos tiempo y dedicar esfuerzo extra a esos temas.

5.3.2. Modo de empleo

La primera vez que se cargue la visualización se mostrará un gráfico de porciones circular con el tiempo dedicado a cada uno de los capítulos publicados del curso. La gráfica es interactiva, y al pulsar sobre uno de los capítulos, su porción se desglosará en 3 porciones como se puede observar en la *Figura 19*. Estas 3 porciones corresponden al tiempo dedicado a contenido evaluable (ejercicios), el tiempo dedicado a contenido no evaluable (teoría) y el tiempo dedicado a la introducción del capítulo. Para volver a mostrar todos los capítulos basta con pulsar sobre cualquier punto de la gráfica.

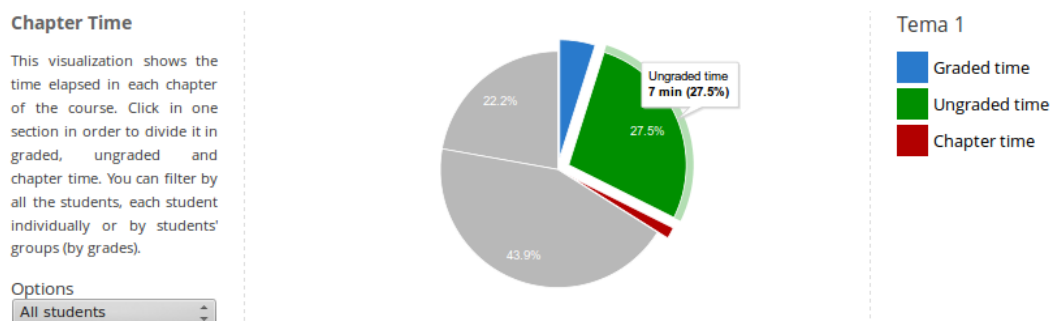


Figura 19: Interacción con la visualización “Chapter Times”

La visualización, también tendrá un selector para seleccionar los datos para toda la clase, por grupos de alumnos según sus calificaciones, o para algún alumno en concreto. Como en el resto de visualizaciones que se muestren a los alumnos, cuando un alumno acceda a ella, solo mostrará sus datos.

5.3.3. Funcionamiento interno

El algoritmo que se emplea para calcular el tiempo calculado en cada uno de los capítulos se basa en recorrer cronológicamente los eventos generados por el usuario almacenados en el *tracking log*. Mientras se recorren estos eventos, se lleva un registro de en que capítulo y *sequential* se encuentra el usuario según el ultimo evento analizado, y cuanto tiempo lleva en el (es decir tiempo inicial en el que accedió al capítulo/*sequential* y el tiempo del último evento analizado). Este registro es el diccionario nombrado *time_data* en el código.

La función que se encarga de *procesar este cálculo* es *get_student_spent_time* y sigue el siguiente proceso:

- Primero inicializa un diccionario que almacena el resultado de tiempo en cada capítulo y en cada *sequential* de cada capítulo, así como inicializa también el diccionario *time_data* explicado anteriormente.
- Posteriormente obtendrá todos los eventos relevantes para el curso y usuario dado, mediante la función del API de datos, *get_course_events_sql*.
- Una vez obtenidos estos datos, recorre todos los eventos, ordenados por orden cronológico, y dependiendo de si el origen de cada evento es el cliente o el servidor, llama a las funciones *manage_browser_event* o *manage_server_event* respectivamente. Estas funciones se encargan de interpretar cada evento, y llamar a las funciones auxiliares *activity_update* o *activity_close*, según consideren si el evento es un evento de mantenimiento de actividad (un cambio de capítulo/*sequential*, o cualquier actividad del

usuario en un capítulo/*sequential*), o de cierre de actividad (*logout*, salida del usuario del courseware, etc). Estas funciones, ayudándose del diccionario de estado, *time_data*, irán añadiendo el tiempo correspondiente a cada capítulo/*sequential* del curso.

- Por último, se llama a la función *activity_close* con el tiempo actual, por si el usuario esta navegando por el curso en el momento que se hace el cálculo de esta estadística.

El funcionamiento de *manage_browser_event* y *manage_server_event* es muy similar, simplemente se diferencian en que los eventos de un origen u otro tienen distinta forma. El proceso general que siguen estas dos funciones se describe en la Figura 20.

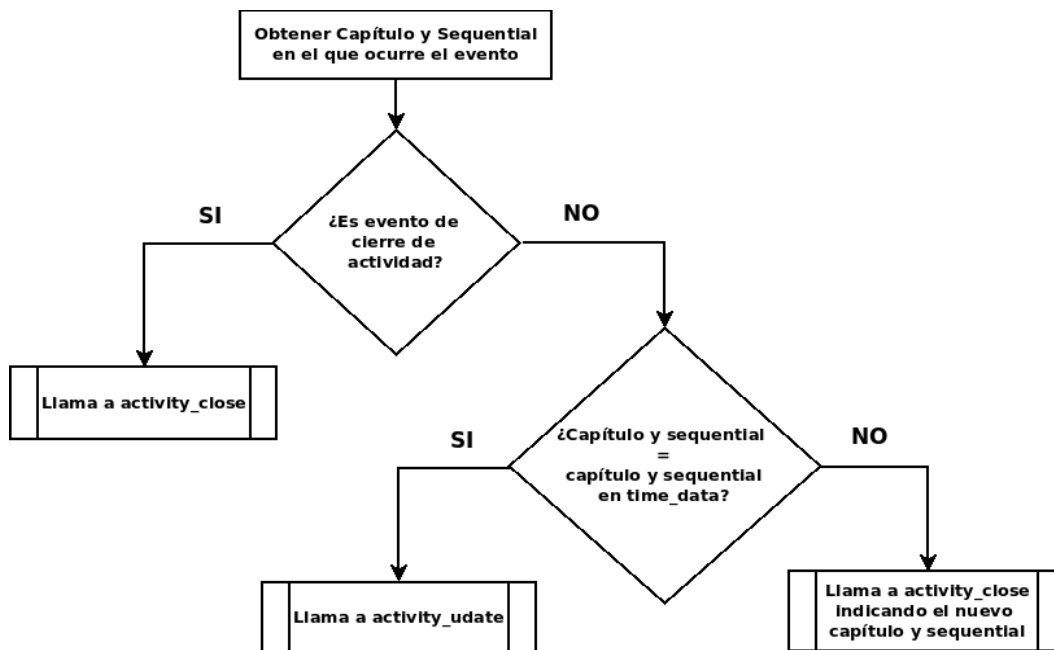


Figura 20: Diagrama de flujo de las funciones *manage_browser_event* y *manage_server_event*

La función *activity_close* se llama cuando ocurre un evento de cierre, o cuando el usuario cambia de capítulo/*sequential*. Calcula el tiempo que ha pasado desde el último evento almacenado en *time_data* hasta el tiempo del nuevo evento, y lo añade al diccionario que almacena los tiempos, mediante la función *add_course_time*. Notar que para evitar posibles contaminaciones de cálculos en situaciones como que el usuario deje abierta la ventana pero no haga uso de ella, se incluye un tiempo de inactividad fijado en la variable global de *analytics.py*, *INACTIVITY_TIME* (actualmente esta fijado a 10 minutos). Si el tiempo de inactividad es superado la función *activity_close* considera que se ha cerrado al pasar dicho tiempo de inactividad. Una vez añadidos los tiempos, se actualizará el diccionario *time_data*: en caso de que el evento sea de cierre, todos sus campos se dejarán vacíos, y en el caso de que sea un cambio de capítulo/*sequential*, *time_data* se actualiza con la nueva localización, así como con el tiempo del evento.

En el caso de *activity_update*, comprueba que no se haya excedido el tiempo de inactividad y simplemente actualiza el campo de *time_data* que indica el tiempo del último evento, al tiempo indicado en el nuevo evento. En caso de que el tiempo de inactividad se haya excedido, llama a la función *add_course_time*, y actualiza los tiempos de *time_data* al tiempo del evento.

Por último, al igual que en el resto de visualizaciones, la función *update_DB_course_spent_time* se encarga de calcular los tiempos de todos los alumnos del curso dado y añadirlo en la tabla de la base de datos creada a partir del modelo *CourseTime*. Los campos de la tabla de la base de datos creada con este modelo, se describen en la Tabla 27.

Tabla 27: Tabla de la base de datos *CourseTime*

Columna	Tipo	Null	Key	Descripción
<i>id</i>	Int	NO	PRI	ID de la entrada en la tabla
<i>student_id</i>	Int	NO		ID del estudiante. Este campo, junto con <i>course_id</i> tienen que ser únicos. Habrá 4 constantes negativas que se emplearán como identificadores de grupos de alumnos: -1: Todos los estudiantes -2: Estudiantes con nota excepcional -3: Estudiantes aprobados -4: Estudiantes suspensos
<i>course_id</i>	CourseKey	NO	MUL	ID del curso. Este campo, junto con <i>student_id</i> tienen que ser únicos.
<i>time_spent</i>	Char	NO		Diccionario con los tiempos dedicados a cada capítulo y <i>sequential</i> . Las claves para dicho diccionario son las IDS del capítulo que se quiere consultar y las entradas tienen la siguiente forma: “ID”:{ “time_spent”: Tiempo dedicado al capítulo, “sequentials”:{} } El campo <i>sequentials</i> incluye diccionarios de igual forma pero con las IDs del <i>sequential</i> como clave.
<i>last_calc</i>	DateTime	NO		Fecha de la última modificación de la entrada

5.4. CHAPTER ACCESES

La visualización, perteneciente a las visualizaciones de actividad en el curso, muestra un gráfico de barras con todos los capítulos del curso, indicando el número medio de accesos de los estudiantes a cada uno de ellos (*Figura 21*). La gráfica es interactiva y permite navegar en los capítulos para ver los accesos desglosados en sus *sequentials* y también desglosar los *sequentials* en sus *verticals*.

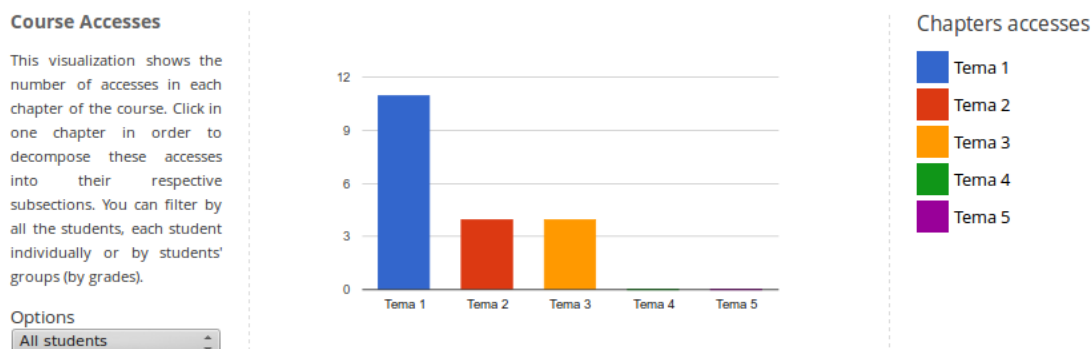


Figura 21: Visualización "Chapter Accesses"

Esta visualización también estará disponible para los alumnos, aunque como en el resto de visualizaciones, solo podrán consultar sus propios datos, e incluye un selector para seleccionar los datos por grupos de alumnos o alumnos concretos por parte de los instructores.

5.4.1. Motivaciones y utilidad

De manera similar a la anterior visualización, esta gráfica permite al instructor observar que capítulos, *sequentials* y *verticals*, están teniendo más dificultad para los alumnos. A diferencia de la anterior, que indicaba el tiempo dedicado en cada sección, en esta gráfica se indica el número de accesos. Con estos datos se puede interpretar que secciones vuelven a consultar más los alumnos, indicando que son secciones con más importancia teórica para el resto de secciones, o de una dificultad extra para memorizar y aprender esta teoría.

Una vez interpretados estos datos, los instructores pueden incluir recordatorios en otras secciones, o realizar más hincapié en los conceptos de la sección para futuros lanzamientos del curso, así como incluir apoyo extra para dichos conceptos.

Los alumnos podrán consultar y observar que secciones están consultando más e intentar reforzar esos conceptos con el objetivo de no tener que volver a consultarlos tanto.

5.4.2. Modo de empleo

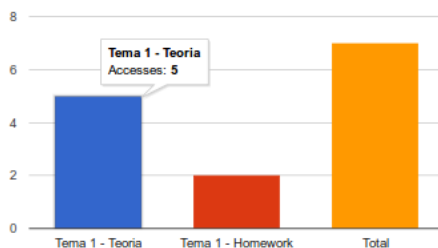
En la primera carga la visualización muestra en un diagrama de barras todos los capítulos publicados del curso, indicando cuantos accesos de media ha tenido cada uno de ellos.

Course Accesses

This visualization shows the number of accesses in each chapter of the course. Click in one chapter in order to decompose these accesses into their respective subsections. You can filter by all the students, each student individually or by students' groups (by grades).

Options

All students



Sequential

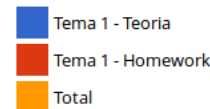


Figura 22: Interacción con la visualización "Chapter Accesses" (Sequentials)

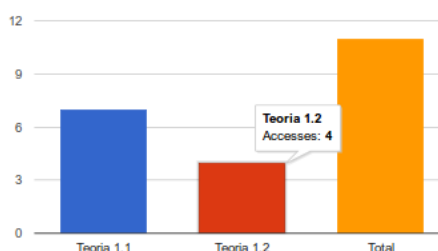
Si se quiere conocer los accesos a cada una de los *sequentials* de algún capítulo basta con pulsar sobre la barra de dicho capítulo para que aparezcan sus *sequentials* desglosados (Figura 22). El funcionamiento es el mismo para desglosar los *verticals* de uno de los *sequentials* (Figura 23). Pulsando sobre la barra que indica el total de los *sequentials* o *verticals* se vuelve a mostrar la capa anterior, la de los *sequentials* en el caso de que se estén mostrando los *verticals*, o a los capítulos en caso de que se estén mostrando los *sequentials*.

Course Accesses

This visualization shows the number of accesses in each chapter of the course. Click in one chapter in order to decompose these accesses into their respective subsections. You can filter by all the students, each student individually or by students' groups (by grades).

Options

All students



Vertical

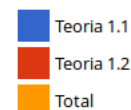


Figura 23: Interacción con la visualización "Chapter Accesses" (Verticals)

Por último la visualización también tiene un selector para los instructores, con el que podrán seleccionar los datos para todos los alumnos, para grupos de alumnos según calificación y para alumnos concretos.

5.4.3. Funcionamiento interno

La función `get_student_section_accesses` calcula, dados un usuario y un curso como parámetros, los accesos de este usuario a los capítulos, *sequentials* y *verticals* de dicho curso. El proceso que sigue la función es el siguiente:

- Se llama a la función auxiliar `create_access_chapters` para crear el diccionario que contendrá la estructura de capítulos, *sequentials* y *verticals* y el número de accesos a cada uno de ellos. Las claves del diccionario son las IDs de cada uno de los capítulos, y la forma de cada entrada es:

```

“ID de capítulo” :
{
  “accesses”: Número de accesos al capítulo,
  “sequentials”: { “ID de sequential”:
    {
      “accesses”: Número de accesos al sequential,
      “last_vert”: Último vertical que se ha cargado
                    en este sequential,
      “verticals”: { “ID de vertical”:
        {
          “accesses”: Número de
                    accesos al vertical
        }
      }
    }
  }
}

```

Como se puede observar, es necesario almacenar cual es el último *vertical* cargado en cada *sequential*, ya que al seleccionar un *sequential* el sistema carga el último *vertical* que el usuario haya visitado dentro de ese *sequential* y no se indica de ninguna forma en el evento.

- Obtiene la lista de eventos relevantes mediante la función `get_course_access_events_sql` del API de datos.
- Inicializa las variables a *None* las variables que almacenaran el capítulo, *sequential* y *vertical* en el que se encuentra el usuario antes de analizar el siguiente evento.
- Recorre todos los eventos añadiendo los accesos a cada capítulo, *sequential* y *vertical* al diccionario creado anteriormente, mediante la función `add_course_access`. Los eventos serán distintos si provienen del servidor o del cliente:
- Eventos del servidor: Estos eventos se generan cada vez que el usuario accede a un nuevo capítulo o *sequential* (los eventos que se generan al cambiar de vertical dentro de un *sequential* se ignoran, ya que pueden dar lugar a equivocación, y es más fiable emplear los que se generan en el cliente). La función obtiene el capítulo y *sequential* mediante `get_locations_from_url` y añade los accesos al diccionario.
- Eventos en el cliente: De los eventos generados en el cliente, solo se interpretaran los que contengan como tipo de evento *seq_prev*, *seq_next* o *seq_goto*, es decir, eventos que se generan cuando el usuario navega entre los *verticals* de un *sequential* ya cargado. Por tanto, si el evento es generado en el cliente, simplemente se añadirá un acceso al vertical indicado en el evento (si este ha cambiado con respecto al anterior vertical cargado).

Como en el resto de visualizaciones, la función `update_DB_course_section_accesses`, se encarga de llamar a la función `get_student_section_accesses`, para cada uno de los alumnos del curso dado, así como para grupos de alumnos y el total de alumnos, y añadir estos datos a la tabla de la base de datos creada a partir del modelo *CourseAccesses*.

Los campos de la tabla de la base de datos que se crea a partir del modelo *CourseAccesses* se muestran en la Tabla 28.

Tabla 28: Tabla de la base de datos CourseAccesses

Columna	Tipo	Null	Key	Descripción
<i>id</i>	Int	NO	PRI	ID de la entrada en la tabla
<i>student_id</i>	Int	NO		ID del estudiante. Este campo junto con <i>course_id</i> tienen que ser únicos. Habrá 4 constantes negativas que se emplearan como identificadores de grupos de alumnos: -1: Todos los estudiantes -2: Estudiantes con nota excepcional -3: Estudiantes aprobados -4: Estudiantes suspensos
<i>course_id</i>	CourseKey	NO	MUL	ID del curso. Este campo junto con <i>student_id</i> tienen que ser únicos.
<i>accesses</i>	Char	NO		Diccionario con los accesos a cada capítulo, <i>sequential</i> y <i>vertical</i> del curso.
<i>last_calc</i>	DateTime	NO		Fecha de la última modificación de la entrada

5.5. PROBLEM & VIDEO PROGRESS

La última visualización que se presenta en esta memoria, se encuentra en la pestaña de visualizaciones de actividad en el curso y muestra el progreso de los estudiantes en la visualización de los vídeos, frente al progreso en los problemas del curso (Figura 24). Para representarlo se emplea una gráfica de área con el progreso de vídeos y la calificación de los problemas frente al paso del tiempo en el curso.

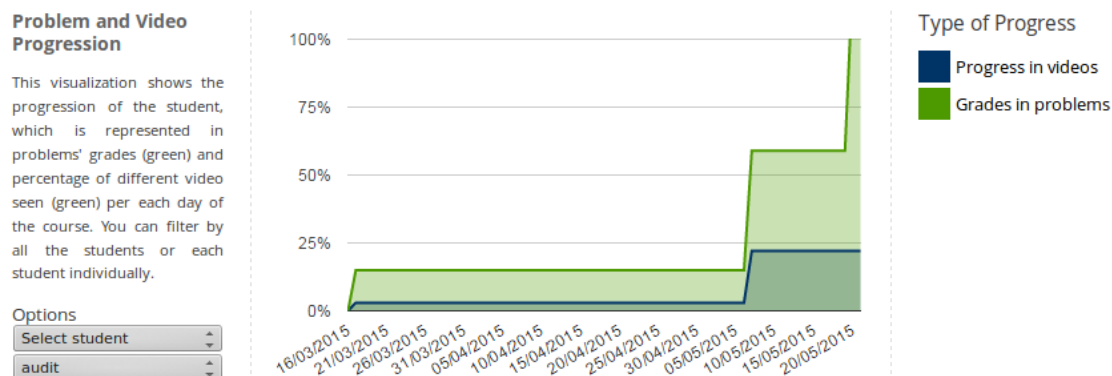


Figura 24: Visualización "Problem & Video Progress"

Los estudiantes podrán consultar esta gráfica con sus propios datos, y la visualización incluye un selector para que los instructores seleccionen los datos de estudiantes concretos, grupos de estudiantes, o de todos los estudiantes.

5.5.1. Motivaciones y utilidad

En la mayoría de cursos de edX, los vídeos se emplean como principal fuente de teoría por parte de los instructores. Por otra parte los problemas y ejercicios, son los elementos que marcan la calificación de los alumnos en el curso. Con esta gráfica se muestra en un rápido vistazo, cual es la evolución de los alumnos a lo largo del periodo de funcionamiento del curso, comparando su progreso en el estudio de la parte teórica y en la parte evaluable del curso.

Gracias a esta gráfica, los instructores podrán observar la evolución de los alumnos a lo largo del periodo que el curso esta activo. También pueden observar carencias o casos fuera de la norma, como por ejemplo que los alumnos saquen buenas notas pero no vean todos los vídeos al completo, lo cual puede significar un exceso de teoría. El caso contrario también puede aparecer: que los alumnos lleven al día la teoría pero no estén obteniendo buenas calificaciones en los ejercicios evaluables.

Los alumnos también pueden consultar como esta siendo su progreso, y observar si necesitan refuerzo en algún punto.

5.5.2. Modo de empleo

El uso de esta visualización no tiene prácticamente ninguna complicación. La gráfica no es interactiva (simplemente muestra los porcentajes al pulsar sobre alguna de las dos áreas) y solo contiene las dos áreas representando el porcentaje de vídeo visto, y el porcentaje de las calificaciones de los alumnos sobre la máxima calificación posible.

Los alumnos podrán consultar sus propios datos, y los instructores tienen un selector para seleccionar los datos de todos los alumnos, de grupos de alumnos según calificación y de alumnos concretos.

5.5.3. Funcionamiento interno

En esta visualización se calcula por separado el progreso en los problemas y en los vídeos.

Para el cálculo del progreso en las calificaciones de los problemas, se emplea la función *get_student_problem_progress*. El proceso que sigue esta función para el cálculo de la estadística es el siguiente:

- Llama a la función *create_course_progress* que devuelve una estructura con todos los problemas del curso calificables y un eje de tiempos calculado a que va desde el inicio

del curso hasta el final de este (o la fecha en que se haga el cálculo si el curso no ha finalizado aún), con una separación de un día.

- Crea el array *problem_progress* que almacenará el resultado de la estadística en forma de duplas conteniendo el porcentaje de puntos obtenidos sobre el total posible y la fecha.
- Obtiene todos los eventos del *tracking log* relacionados con calificaciones en problemas llamando a la función *get_problem_history_sql* del API de datos.
- Recorre el eje de tiempos, filtrando en cada iteración los eventos ocurridos entre la fecha de la iteración y la anterior. A partir de los eventos de tipo *problem_check* y *problem_rescore* añade o cambia en la estructura creada en el primer paso las notas de los problemas. Una vez actualizada la estructura con todos los eventos que ocurran entre las dos fechas, recalcula el porcentaje total de puntos y lo añade al array *problem_progress* junto con la fecha.
- Devuelve el array *problem_progress* conteniendo el porcentaje de la calificación del alumno con respecto al total posible en cada una de las fechas del eje de tiempo.

La función *update_DB_course_problem_progress* llama a la función *get_student_problem_progress* para todos los alumnos y calcula la media para grupos de alumnos y para el total de los alumnos ayudándose de las funciones auxiliares *add_problem_progress* y *mean_problem_progress_sum*. Una vez calculadas todas las estadísticas las añade a la tabla de la base de datos creada a partir del modelo *CourseProbVidProgress*.

Para el caso del progreso de los alumnos en la visualización de los vídeos, se emplea la función *get_student_video_progress*. El proceso del cálculo es el siguiente:

- Llama a la función *create_course_progress* para obtener un eje de tiempos que va desde el inicio del curso hasta el final de este (o la fecha en que se haga el cálculo si el curso no ha finalizado aún), con una separación de un día.
- Inicializa el array *video_progress* que almacenará duplas de valores indicando una fecha y el porcentaje de visualización de vídeos en esa fecha. También llama a las funciones *get_info_videos* y *get_video_events_interval*, diseñadas por Héctor Javier Pijeira Díaz para el API de datos, que devolverán la lista con las IDs y duraciones de los vídeos, así como el tiempo en que ocurre el primer y el último evento relacionado con vídeos generados por el usuario.
- Recorre el eje de tiempos, añadiendo en cada iteración una dupla con el porcentaje de vídeo visualizado y el tiempo al array *video_progress*. Para calcular el porcentaje de vídeos visualizados por el usuario en cada intervalo de tiempo se emplea la función *student_total_video_percent*, que es una adaptación de la función *video_consumption*, diseñada por Héctor Javier Pijeira Díaz para el API de datos.

Al igual que en el caso del progreso de problemas, la función *update_DB_course_video_progress* se encarga de calcular las estadísticas para todos los usuarios, así como para grupos de usuarios, ayudándose de diversas funciones auxiliares. Una vez calculadas y optimizadas las estadísticas, las añade a la tabla de la base de datos creada a partir del modelo *CourseProbVidProgress*.

Los campos de la tabla de la base de datos creada a partir del modelo *CourseProbVidProgress* se muestran en la Tabla 29.

Tabla 29: Tabla de la base de datos *CourseProbVidProgress*

Columna	Tipo	Null	Key	Descripción
<i>id</i>	Int	NO	PRI	ID de la entrada en la tabla
<i>student_id</i>	Int	NO		ID del estudiante. Este campo, junto con <i>course_id</i> tienen que ser únicos. Habrá 4 constantes negativas que se emplearan como identificadores de grupos de alumnos: -1: Todos los estudiantes -2: Estudiantes con nota excepcional -3: Estudiantes aprobados -4: Estudiantes suspensos
<i>course_id</i>	CourseKey	NO	MUL	ID del curso. Este campo, junto con <i>student_id</i> tienen que ser únicos.
<i>progress</i>	Char	NO		Array en forma de texto con los porcentajes sobre 100 del progreso
<i>type</i>	Char	NO		Tipo del progreso. Los posibles valores son <i>PROB</i> para problemas y <i>VID</i> para vídeos.
<i>start_time</i>	DateTime	YES		Fecha de inicio de los valores almacenados en el campo <i>progress</i>
<i>end_time</i>	DateTime	YES		Ultima fecha de los valores almacenados en el campo <i>progress</i>
<i>delta</i>	Float	NO		Separación entre muestras en segundos
<i>last_calc</i>	DateTime	NO		Fecha de la última modificación de la entrada

CAPÍTULO 6

6. CONCLUSIONES, RESULTADOS Y TRABAJOS FUTUROS

En este último capítulo se abordan, las conclusiones, resultados y trabajos futuros relacionados con este proyecto, que forma parte de un esfuerzo común, en el que hay varios trabajos de distintas personas, algunos ya realizados, y otros en curso, que abordan distintas partes del módulo de *learning analytics* desarrollado.

Se analizan, por una parte, las conclusiones finales del proyecto presentado en esta memoria, indicando si se han cumplido los objetivos planteados al principio de esta. Por otra parte, se presentan algunos de los resultados obtenidos con la presentación de este módulo a la comunidad, como por ejemplo diversas publicaciones o evaluaciones realizadas sobre la funcionalidad de este módulo. Por último, se presentan algunos de los posibles trabajos futuros que mejorarían y potenciarían el uso y la utilidad de este módulo.

6.1. CONCLUSIONES

A continuación se analiza el grado de cumplimiento de cada uno de los 4 objetivos planteados al comienzo de esta memoria, presentando los resultados obtenidos y los diversos problemas encontrados a la hora de alcanzarlos:

6.1.1. Familiarización con la plataforma Open edX y las tecnologías empleadas para trabajar en ella

Como se comenta en el Capítulo 1, el primer objetivo de este proyecto, es familiarizarse con la arquitectura de Open edX y con las tecnologías que se emplean para desarrollar el módulo de *learning analytics*.

El análisis de la estructura que sigue el proyecto Open edX se ha detallado en la primera sección del Capítulo 3. Se puede concluir que el análisis ha sido satisfactorio, presentando una idea general de la arquitectura de la plataforma, y concretamente de las partes que interesan para el desarrollo del módulo de *learning analytics*. Debido a la temprana edad del proyecto, no se han encontrado estudios similares, por lo que la definición de la arquitectura de Open edX puede ser de gran utilidad para la comunidad, y en especial para personas que entren en contacto con el desarrollo en la plataforma por primera vez. Por la misma razón, los principales problemas que se han encontrado para el desarrollo del análisis de esta arquitectura, han sido la falta de documentación de muchas de las partes, teniendo que analizar el proyecto a nivel de código, con lenguajes y tecnologías desconocidas previamente en muchos casos.

Por otra parte, y como condición para poder analizar correctamente la arquitectura, ha sido necesaria una previa familiarización con muchos de los lenguajes y tecnologías empleados para el desarrollo de la plataforma Open edX, que ha supuesto un empleo de tiempo considerable.

6.1.2. Diseñar un “API” para la obtención de datos de la plataforma Open edX

El diseño de un API para obtener datos de la plataforma de forma sencilla es uno de los objetivos más importantes de este proyecto. Por una parte simplifica en gran medida el desarrollo del módulo, encapsulando la obtención de datos de la plataforma Open edX, que puede resultar tediosa en algunas situaciones. También hace más sencillo el mantenimiento del módulo ante cambios en la forma de obtener datos de la plataforma, donde solo habrá que cambiar el código del API y el resto de partes del módulo seguirán funcionando correctamente. Por último, aunque no menos importante, ofrece a la comunidad una funcionalidad que puede ser aprovechada por cualquier desarrollador que quiera obtener datos de la plataforma.

Se puede concluir que el desarrollo del API ha sido satisfactorio, incluyendo gran número de funciones, que junto con las incluidas por otros miembros del equipo que trabaja en el módulo de *learning analytics*, ofrece un API base con gran número de funciones muy útiles para la obtención de diversos tipos de datos. La descripción de las funciones desarrolladas en este proyecto se encuentra en el Capítulo 4 de esta memoria.

Además de la utilidad que ofrece para el desarrollo de este proyecto y para la comunidad, el otro objetivo del diseño de esta API, es que una vez publicada pueda ser ampliada y mantenida por la propia comunidad de Open edX, aunque debido a la reciente publicación del módulo, aún no se ha podido comprobar el impacto que este API ha tenido en la comunidad de desarrolladores.

6.1.3. Desarrollar un conjunto de visualizaciones de analítica del aprendizaje para Open edX

El objetivo principal de este proyecto era desarrollar un módulo de *learning analytics* que analice los datos de los cursos de la plataforma Open edX y muestre estadísticas a los instructores y alumnos, en forma de visualizaciones, con el objetivo de mejorar el proceso educativo del curso.

En el Capítulo 3 de esta memoria se describe la arquitectura de este módulo, y en el Capítulo 5 se presentan las visualizaciones que se han incluido como parte de este proyecto. Podemos concluir que el objetivo planteado se ha cumplido satisfactoriamente, aunque algunas de las visualizaciones presentan diversos problemas que escapan a los objetivos del proyecto. Estos problemas son principalmente problemas de escalabilidad en los cálculos de las estadísticas, la necesidad de incluir un sistema de pruebas automáticas y la falta de visualizaciones de más alto nivel. Todos estos problemas se tratarán con más detalle en la sección «Trabajos futuros» de este capítulo.

6.1.4. Propuesta e implementación de arquitectura para integrar el modulo de analítica del aprendizaje en Open edX

Para cumplir el último objetivo, integrar el modulo de *learning analytics* en la plataforma Open edX, se ha trabajado conjuntamente con otros miembros del equipo. Se han integrado las visualizaciones presentadas en el Capítulo 5 de esta memoria y las desarrolladas por los miembros del equipo, dando lugar a un módulo de *learning analytics*, compuesto por 12 visualizaciones distintas, que recibe el nombre de ANALYSE. [30]

Actualmente ANALYSE está publicado para la comunidad bajo licencia AGPL [31] y se encuentra alojado en GitHub [32]. El módulo esta publicado como *beta*, y actualmente se encuentra en proceso de desarrollo, aunque cualquier persona puede instalarlo, ya sea para uso o testeo, o para colaborar en el desarrollo. Las mejoras mas necesarias en el desarrollo del módulo se describen en la sección «Trabajos futuros» de este capítulo.

Por tanto, podemos concluir que los objetivos presentados al inicio de este proyecto, se han cumplido satisfactoriamente, dado que el objetivo de este proyecto no era construir un modulo de *learning analytics* que cubriera todos los aspectos necesarios, sino realizar una primera aproximación, que junto con el trabajo de otras personas acabe dando lugar a un módulo más escalable, estable y completo.

6.2. PRUEBAS Y RESULTADOS

En esta sección, por una parte se presentan los resultados de una evaluación el módulo realizada en una clase de la Universidad Carlos III de Madrid, y por otra parte se listaran algunos de los *papers* que tratan sobre el módulo ANALYSE presentados a diversas conferencias y revistas, realizados tanto durante el desarrollo del módulo, como después de la publicación de este.

6.2.1. Evaluación de ANALYSE

Fuera del marco de este proyecto fin de carrera, junto con otros miembros del equipo del módulo de *learning analytics*, se ha realizado una presentación y una evaluación del módulo ANALYSE ante alumnos de la Universidad Carlos III de Madrid. Los participantes en la evaluación fueron 40 estudiantes del Máster de Ingeniería de Telecomunicación que cursaban la asignatura “Diseño de Aplicaciones Telemáticas” de dicha Universidad.

La metodología de la evaluación fue permitir a los participantes interactuar brevemente con un curso diseñado en Open edX y con el módulo ANALYSE, para posteriormente realizarles una serie de preguntas con el objetivo de evaluar la usabilidad, la utilidad y la efectividad tanto del módulo ANALYSE en general, como de cada una de las visualizaciones individualmente.

Los resultados de la evaluación fueron muy positivos en la mayoría de los puntos, y se separan en tres categorías: usabilidad, utilidad y efectividad. Como conclusión, se puedo

observar que el módulo ANALYSE muestra un potencial muy grande, siendo sencillo de usar y ofreciendo una gran utilidad a los instructores, por lo que es importante seguir trabajando en el con el objetivo de conseguir un módulo más completo y que se pueda integrar fácilmente en la rama oficial de la plataforma Open edX.

6.2.2. Publicaciones presentadas a la comunidad

Durante el desarrollo del módulo ANALYSE y posteriormente a su publicación, se han escrito una serie de artículos por parte del equipo de desarrollo, con el objetivo de que fueran publicados en varias conferencias y revistas, de forma que ANALYSE se diera a conocer a la comunidad.

- **“Towards the development of a learning analytics extension in open edX”. Javier Santofimia Ruiz, Héctor J. Pijeira Díaz, José A. Ruipérez-Valiente, Pedro J. Muñoz-Merino, Carlos Delgado Kloos. TEEM Conference (2014) [33]**

Este fue primer paper que se escribió sobre el módulo de *learning analytics* para Open edX (que aún no estaba nombrado como ANALYSE) y el objetivo era presentarlo en las conferencias TEEM (Technological Ecosystems for Enhancing Multiculturality) [34] que tuvo lugar en el Instituto Universitario de Ciencias de la Educación de la Universidad de Salamanca (IUCE).

En el paper se hace un análisis la situación en aquel momento de la aplicación de métodos de *learning analytics* en la plataforma Open edX, así como una primera aproximación a la arquitectura de la plataforma. Por último se presentan los primeros pasos dados para el diseño del módulo de *learning analytics* para Open edX, que posteriormente recibiría el nombre de ANALYSE.

- **“Using Video Visualizations in Open edX to Understand Learning Interactions of Students”. Héctor J. Pijeira Díaz, Javier Santofimia Ruiz, José A. Ruipérez-Valiente, Pedro J. Muñoz-Merino, Carlos Delgado Kloos. EC-TEL (2015) [35]**

Se trata de la introducción a una demo para la conferencia EC-TEL (European Conference on Technology Enhanced Learning) que tendrá lugar en Toledo en Septiembre de 2015. En esta demo se mostrará el funcionamiento del módulo ANALYSE, centrándose en las visualizaciones que permiten detectar problemas en vídeos del curso.

- **“Combining Learning Analytics and the Flipped Classroom in a MOOC of maths”. Diego Redondo Martínez, Pedro J. Muñoz-Merino, José A. Ruipérez-Valiente, Héctor J. Pijeira Díaz, Javier Santofimia Ruiz, Carlos Delgado Kloos. WAPLA (EC-TEL) (2015) [36]**

Este paper, relata la experiencia de la aplicación de un MOOC mediante la plataforma Open edX y del uso del módulo ANALYSE en un curso de matemáticas del Centro Comarcal de Educación de Personas Adultas Sierra Norte de Torrelaguna (Madrid).

En el artículo, se relata tanto el proceso de creación de un curso de matemáticas en la plataforma Open edX, como la forma de aprovechar las diferentes visualizaciones del módulo ANALYSE.

El paper ha sido aceptado para el taller WAPLA (Workshop on Applied and Practical Learning Analytics) dentro de la conferencia EC-TEL citada anteriormente.

- **“Design and Evaluation of a Learning Analytics Application for Open edX”.** José A. Ruipérez-Valiente, Pedro J. Muñoz-Merino, Héctor J. Pijeira Díaz, Javier Santofimia Ruiz, Carlos Delgado Kloos. Pending [37]

Este artículo tiene objetivo presentar el módulo ANALYSE a la comunidad, ha sido enviado al Journal of IEEE Transactions on Human-Machine Systems y aun se encuentra en proceso de aceptación.

El artículo es algo más extenso que los anteriores y en el se presenta todo el proceso de diseño del módulo ANALYSE. Comienza con un análisis de la situación de Open edX en cuanto a la aplicación de *learning analytics*. Posteriormente se presenta la arquitectura del módulo ANALYSE dentro de la plataforma, y una descripción de cada una de las visualizaciones incluidas en el módulo hasta el momento. Por último, se incluyen los datos sobre la evaluación de ANALYSE presentada en la sección anterior.

En general, se puede concluir, que el módulo ANALYSE, a pesar de su temprano estado desarrollo, ya esta ofreciendo resultados positivos, y una buena aceptación por parte de la comunidad, lo cual es una buena razón para continuar su desarrollo y mejora de cara al futuro.

6.3. TRABAJOS FUTUROS

A pesar de que el módulo ya esta publicado, este se encuentra aún en fase beta, y queda mucho trabajo por hacer, ya sea por los propios desarrolladores actuales, o por nuevos desarrolladores. En esta sección se hace un análisis de las posibles mejoras a implementar en un futuro para completar el módulo de *learning analytics* ANALYSE:

6.3.1. Escalabilidad

Uno de los principales problemas que presenta el módulo publicado, es la alta carga de trabajo que supone calcular todas las estadísticas para un gran numero de alumnos. El módulo ha sido diseñado y probado bajo un supuesto de cursos tradicionales, es decir, con pocos alumnos. Aunque se ha intentado que los cálculos de las estadísticas consuman el mínimo posible y se emplea un encolador de tareas periódicas (Celerybeat) para no

sobrecargar el sistema, en el caso de que ANALYSE se empleara en un despliegue con miles de alumnos y múltiples cursos, supondría una carga muy considerable.

Por tanto, uno de las mejoras más importantes a realizar en el módulo de *learning analytics*, es solucionar estos problemas de escalabilidad que presenta. La solución a estos problemas pasa por mejorar la eficiencia de las funciones que calculan las estadísticas. Una primera aproximación podría ser emplear los datos obtenidos en el último cálculo como base para calcular los nuevos valores. Esto no es complicado en muchas de las visualizaciones, ya que la mayoría almacenan la fecha en la que se hizo el último cálculo, aunque en ciertas visualizaciones (como las que dependan de obtener notas de los alumnos) puede tener una aplicación más complicada. Otro enfoque que puede resultar positivo, es mejorar la eficiencia de la obtención de datos directos de la plataforma, aunque esto puede requerir ser desarrollado por personas con un conocimiento más profundo de la arquitectura y funcionamiento de la plataforma Open edX.

Ya existe un proyecto dedicado a la mejora de eficiencia de las visualizaciones dentro del grupo de trabajo de ANALYSE, pero es un elemento que necesitará estar en constante desarrollo. En general estas mejoras requerirán trabajo, y posiblemente, como se ha comentado, la ayuda de desarrolladores con un mayor conocimiento de la plataforma, pero son mejoras que hay que realizar obligatoriamente si se quiere emplear en una plataforma funcionando con el objetivo de Open edX, es decir, como curso MOOC.

6.3.2. Nuevas visualizaciones

En el desarrollo del módulo ANALYSE se han intentado incluir visualizaciones que cubran gran parte de los datos que ofrece la plataforma Open edX. A pesar de esto, y debido a que el módulo se encuentra en una fase temprana de desarrollo, hay muchos temas que podrían resultar útiles para los usuarios que no se cubren con las visualizaciones actuales. Asimismo muchas de las visualizaciones incluidas pueden necesitar una interpretación por parte de los usuarios que podía ser sustituida por visualizaciones de nivel más alto.

Por tanto, por una parte sería positivo que se desarrollaran nuevas visualizaciones con temas que no cubren las actuales, como el uso de los foros de discusión, o la participación de los alumnos en los ejercicios con revisión por pares. Por otra parte, la inclusión de visualizaciones que implementen sistemas de inteligencia artificial, capaces de analizar los datos de la plataforma (nuevos datos, o los propios datos tratados por otras visualizaciones) para sacar conclusiones y alertas, podría quitar una gran carga de trabajo de análisis a los usuarios del módulo.

Dentro del grupo de desarrollo de ANALYSE existen varios proyectos con el objetivo de crear nuevas visualizaciones y mejorar las actuales.

6.3.3. Mantenimiento y soporte de bugs

Por la naturaleza del módulo (proyecto para una plataforma abierta en constante cambio), será necesario un mantenimiento activo del código por diversos motivos.

Por una parte, la plataforma Open edX se encuentra en constante desarrollo, por lo que un cambio importante en ella puede llevar a que el módulo ANALYSE deje de funcionar correctamente (aunque actualmente se encuentra integrado en una *release* específica de la plataforma, en un futuro debería ser independiente a esta y funcionar con cualquier *release*). Gracias a la modularidad que ofrece el API de datos diseñado para el módulo, los cambios en la obtención de datos no supondrán un cambio drástico del código del módulo, pero aún así, será necesario actualizar el API de forma que la obtención de datos funcione correctamente.

Otra parte importante del mantenimiento será el soporte de bugs que se vayan encontrando conforme el módulo empieza a ser probado. Para encontrar estos bugs, sería muy positivo diseñar un sistema de pruebas automáticas, que asegure que la mayoría de bugs existentes en el código sean localizados.

6.3.4. Mejora de integración y sencillez de instalación

Como se ha comentado anteriormente, actualmente el módulo está publicado junto a un *release* concreto de la plataforma Open edX, de forma que no haya problemas de compatibilidad ante cambios en la plataforma. Por tanto, la única forma de instalar el módulo, es instalar toda la plataforma que se encuentra en el repositorio de ANALYSE junto con el módulo de *learning analytics*.

En un futuro, sería positivo trabajar para hacer independiente el módulo del *release* en el que se encuentre la plataforma Open edX, de forma que cualquier persona con la plataforma instalada desde el repositorio oficial, pueda integrar el módulo de manera sencilla y sin tener que instalar el *release* la plataforma al completo para la que está diseñado ANALYSE. Esto supondría un extra de trabajo para el mantenimiento del módulo, por lo que el trabajo comentado en el punto anterior deberá estar correctamente organizado, antes de ofrecer la instalación del módulo sobre cualquier *release* de la plataforma Open edX.

Como paso final, sería interesante, trabajando junto con el equipo de desarrollo de Open edX, integrar el módulo ANALYSE en el repositorio oficial de la plataforma, de forma que cualquier persona que instale la plataforma, cuente con las ventajas y utilidades que ofrece este módulo a los usuarios.

7. REFERENCIAS

- [1]. “*Schools and Partners*”. edX.
- [2]. “*The Year of the MOOC*”. Pappano, Laura . The New York Times 2.12 (2012).
- [3]. “*MOOCs: So Many Learners, So Much Potential...*”. Kay, Judy, et al. IEEE Intelligent Systems 3 (2013)
- [4]. “*Verified Certificates of Achievement*”. edX.
- [5]. “*edX FAQs*”. edX.
- [6]. “*Free online AP courses debut on edX Web site*”. The Washington Post (October 2014).
- [7]. “*Stanford to collaborate with edX to develop a free, open source online learning platform*”. Stanford University.
- [8]. “*World’s first MOOC Breaks 3 Million Registered Learners Worldwide*”. ALISON.
- [9]. “*Open Education For A Global Economy*”. Bornstein, David. New York Times (November 2012).
- [10]. “*Stanford Takes Online Schooling To The Next Academic Level*”. All Things Considered, National Public Radio (January 23, 2012).
- [11]. “*Udacity’s Sebastian Thrun, godfather of free online education, changes course*”. Chafkin, Max. Fast Company (20 December 2013).
- [12]. “*Community:Coursera*”. Coursera (November 23, 2013).
- [13]. “*Coursera, the Other Stanford MOOC Startup, Officially Launches with More Poetry Classes, Fewer Robo-Graders*”. Hack Education (April 18, 2012).
- [14]. “*About Khan Academy*”. Khan Academy.
- [15]. “*Does the Khan Academy Pass the MOOC “Duck Test”?*”. EdTech Magazine (April 2013).
- [16]. “*ALAS-KA: A Learning Analytics Extension for Better Understanding the Learning Process in the Khan Academy Platform*”. Ruiperez-Valiente, J.A., Muñoz-Merino, P.J., Leony, D., Delgado Kloos, C.. Computers in Human Behavior (Accepted for publication 2014).
- [17]. “*Miríada X es la segunda plataforma Mooc del mundo y la primera en español...*”. Think Big (Telefónica) (July 2014).
- [18]. “*Cronología de la revolución de la educación*”. Revista Educación Virtual.
- [19]. “*Penetrating the Fog Analytics in Learning and Education*”. George Siemens and Phil Long EDUCAUSE Review Volume 46, Number 5 (September/October 2011).
- [20]. “*Open Learning Analytics: an integrated & modularized platform*”. G. Siemens, D. Gasevic, C. Haythornthwaite, S. Dawson, S. B. Shum, R. Ferguson, E. Duval, K. Verbert, and R. S. J. D. Baker. (2011).
- [21]. “*Academic Analytics*”. Campbell, J.P. and Oblinger, D.G.. EDUCAUSE Quarterly (October 2007).
- [22]. “*The learning analytics cycle: dosing the loop effectively*”. Clow, Doug. (2012).
- [23]. “*MOOCs and the Funnel of Participation*”. Clow, Doug. (2013).
- [24]. “*edX Insights is now available!*”. edX Course Author Support.
- [25]. “*Celery documentation*”. Celery.
- [26]. “*Google Charts Overview*”. Google Developers.
- [27]. “*Django, Documentation:Databases*”. Django.
- [28]. “*Events in the Tracking Log*”. edX Research Guide.
- [29]. “*Using RabbitMQ*”. Celery Documentation.
- [30]. “*ANALYSE*”. UC3M.
- [31]. “*Edx-platform LICENSE file*”. GitHub.
- [32]. “*ANALYSE repository*”. GitHub.
- [33]. “*Towards the development of a learning analytics extension in open edX*”. Javier Santofimia Ruiz, Héctor J. Pijera Díaz, José A. Ruipérez-Valiente, Pedro J. Muñoz-Merino, Carlos Delgado Kloos.

- [34]. “*TEEM (Technological Ecosystems for Enhancing Multiculturality)*”.
- [35]. “*Using Video Visualizations in Open edX to Understand Learning Interactions of Students*”. Héctor J. Pijera Díaz, Javier Santofimia Ruiz, José A. Ruipérez-Valiente, Pedro J. Muñoz-Merino, Carlos Delgado Kloos.
- [36]. “*Combining Learning Analytics and the Flipped Classroom in a MOOC of maths*”. Diego Redondo Martínez, Pedro J. Muñoz-Merino, José A. Ruipérez-Valiente, Héctor J. Pijera Díaz, Javier Santofimia Ruiz, Carlos Delgado Kloos.
- [37]. “*Design and Evaluation of a Learning Analytics Application for Open edX*”. José A. Ruipérez-Valiente, Pedro J. Muñoz-Merino, Héctor J. Pijera Díaz, Javier Santofimia Ruiz, Carlos Delgado Kloos.

